

HỆ THỐNG THÔNG TIN QUẢN LÝ

Khảo sát, Phân tích, Thiết kế hệ thống

Thế giới ý niệm

Tư duy logic để tìm giải pháp

Hệ thống cũ
đang làm gì

Phân tích

Hệ thống mới
Sẽ phải làm gì

Yêu cầu đối với
Hệ thống là gì

Khảo sát

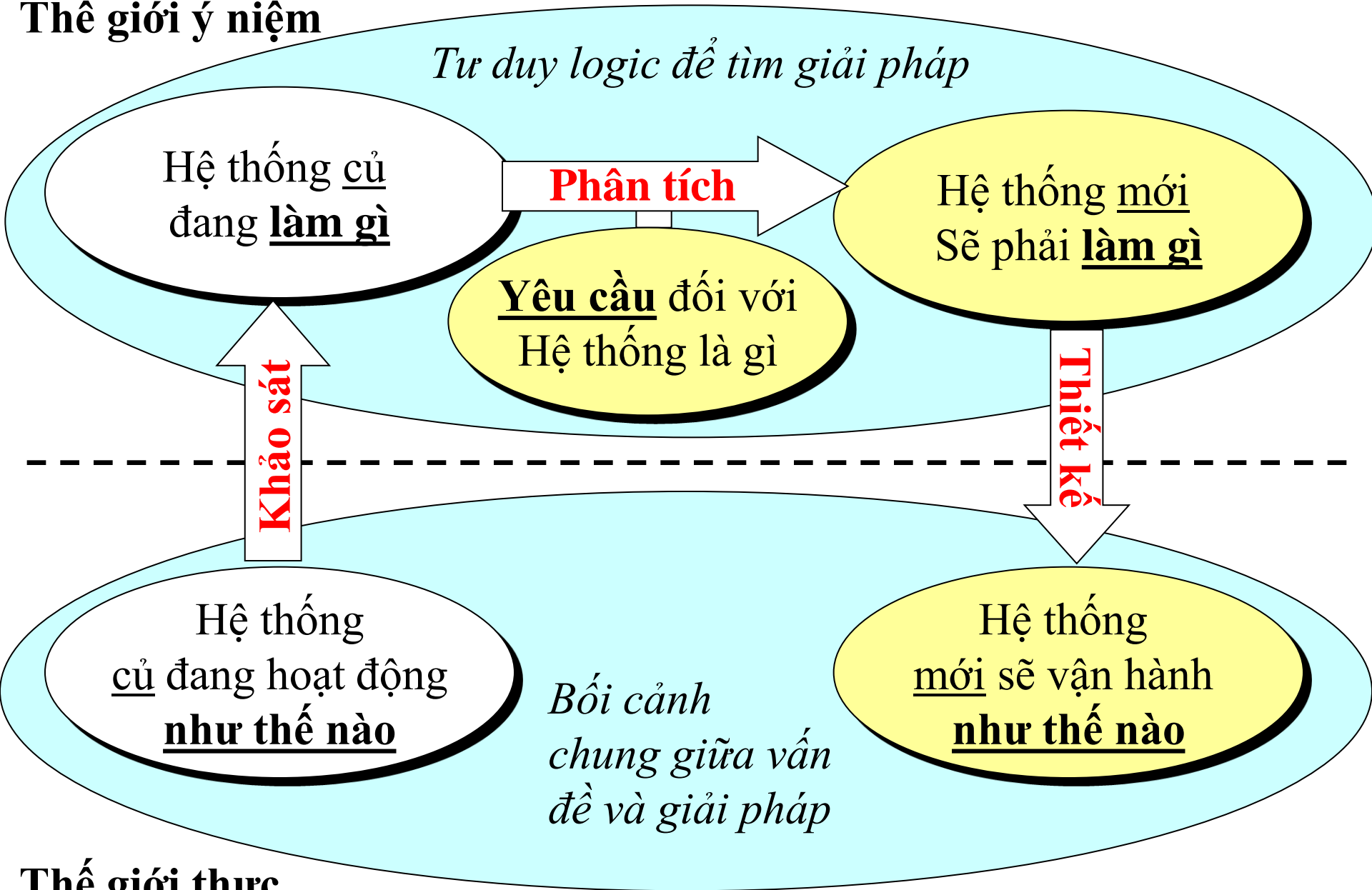
Thiết kế

Hệ thống
cũ đang hoạt động
như thế nào

*Bối cảnh
chung giữa vấn
đề và giải pháp*

Hệ thống
mới sẽ vận hành
như thế nào

Thế giới thực



- ➡ Khảo sát hiện trạng là 1 quá trình khám phá cách mà hệ thống đã được thiết kế và vận hành trong tổ chức, làm bộc lộ các quan hệ nội tại giữa các thành phần trong hệ thống; để từ đó hiểu được hệ thống đang hoạt động như thế nào.
- ➡ Khảo sát hiện trạng là một quá trình tổng hợp thông tin mang tính chất hệ thống, không thể dựa vào lời phát biểu của 1 nhân viên trong tổ chức, vì
 - Mỗi nhân viên chỉ nhìn hệ thống theo một lĩnh vực chuyên môn mà anh ta/ cô ta đang phụ trách, do đó các phát biểu thường không bộc lộ được các ràng buộc tổng thể của hệ thống
 - Các phát biểu của nhiều người thường có mâu thuẫn nhau do mỗi người có cách nhìn khác nhau về hệ thống hiện tại

1. Tìm hiểu tổ chức

- Mục đích, mục tiêu, các kế hoạch ngắn và dài hạn
- Vai trò của hệ thống đang khảo sát trong tổ chức

2. Tìm hiểu các quy trình giữa các bộ phận trong hệ thống

- “Công việc”: quy trình-thủ tục, đầu vào, kết quả
- “Nguồn lực”: khối lượng, phương tiện (facilities), nhân lực

3. Tìm hiểu thông tin – dữ liệu của quy trình

- Quy định, hướng dẫn, tiêu chuẩn
- Dòng dữ liệu, forms/reports (thông tin gì, khi nào, tại sao,...)

4. Hệ thống thông tin quản lý trên máy tính hiện có

- Phạm vi, mức độ và cách nó trợ giúp users thực hiện công việc
- Vai trò (roles) của các users trong hệ thống.
- Phần mềm, mạng máy tính, thiết bị,...

➔ *Các phương pháp Truyền thống*

1. Phỏng vấn cá nhân, nhóm (interviews)
2. Phiếu thăm dò (questionnaires)
3. Quan sát người sử dụng / tham quan thực tế
4. Phân tích tài liệu

Các pp khảo sát truyền thống không tạo cơ hội tốt để kết hợp các loại nguồn lực có sẵn (vd: người sử dụng hoặc giải pháp mới của thế giới) cho việc phát triển hệ thống.

➔ *Các phương pháp hiện đại*

1. “Tương tác” : Prototyping
2. “Cải cách” : Business Process Reengineering (BPR)

➡ Phỏng vấn: tiếp xúc, hỏi vài người để lấy thông tin.

- Phỏng vấn những người nhân viên: Công việc của họ, thông tin mà họ cần để làm việc, cách xử lý thông tin,...
- Phỏng vấn những người quản lý: Xu hướng của tổ chức, các chính sách đang và sẽ áp dụng, mong muốn thay đổi, những ý kiến đánh giá về hệ thống hiện tại,...

➡ Ưu điểm

- Có cơ hội hỏi thêm về những gì vừa mới biết

➡ Khuyết điểm

- Có thể có mâu thuẫn ý kiến riêng giữa các cá nhân
- Tốn nhiều thời gian nếu cần phỏng vấn nhiều người

➡ Phỏng vấn nhiều người chủ chốt cùng một lúc (qua cuộc họp, hội thảo)

➡ Ưu điểm

- Ít tốn thời gian hơn phỏng vấn từng người
- Gia tăng sự trao đổi về các “findings” giữa những người tham gia phỏng vấn
- Hạn chế bớt sự mâu thuẫn ý kiến cá nhân

➡ Nhược điểm: khó thu xếp cho cuộc phỏng vấn

- Do khoảng cách về kiến thức chuyên môn
- Sắp xếp thời điểm và địa điểm họp cho nhiều người cùng một lúc
- Do quan hệ giữa các cá nhân

- ➡ Gửi câu hỏi khảo sát đến nhiều người. Câu hỏi khảo sát phải hết sức rõ ràng, dễ hiểu và dễ trả lời đối với đa số.
- ➡ Ưu điểm
 - Rẻ hơn các loại phỏng vấn, và qua thống kê trên số lượng lớn phiếu thăm dò quay về có thể nhận được thông tin tương đối khách quan.
- ➡ Nhược điểm
 - Không có cơ hội để hỏi thêm !
 - Không chắc chắn ai là tác giả, và mức độ thông tin (trả lời) chính xác đến cỡ nào !!
 - Số phiếu quay về có thể không như mong muốn (quá ít)

So sánh Interviews và Questionnaires

9

Tính chất	Interviews	Questionnaires
Giàu thông tin	Cao	T.bình - Thấp
Thời gian	Có thể rất lâu	Thấp – T.bình
Chi phí	Có thể cao	vừa phải
Tìm hiểu sâu thêm	Tốt	Giới hạn
Độ tin cậy	Cao. Đã biết rõ người được phỏng vấn.	Không cao. Không xác định được tác giả.
Mức độ cộng tác	Người được phỏng vấn cùng tham gia giải quyết vấn đề và cam kết thực hiện	Không rõ các cam kết
Người tham dự	Số lượng giới hạn, đáp ứng tốt	Số lượng lớn, đáp ứng không tốt.

➡ Để biết họ thường làm gì, và ứng xử thế nào cho công việc, đồng thời để đánh giá mức độ hiệu quả của các quy trình và các công cụ hỗ trợ cho các công việc.

➡ Ưu điểm

- Kiểm chứng được công việc thực tế
- Ước lượng được cường độ công việc (workload)

➡ Khuyết điểm

- Sự quan sát có thể không khách quan, do người sử dụng thay đổi thói quen hàng ngày.
- Tốn nhiều thời gian ngồi quan sát.

➡ Phân tích các tài liệu (văn bản) mô tả hệ thống, các tiêu chuẩn, yêu cầu cho hệ thống.

- Tham khảo các văn bản quy trình đang sử dụng.
- Bản thiết kế hệ thống.
- Các mẫu nhập liệu (forms), các báo cáo (reports).

➡ Ưu điểm:

- Có nhiều thông tin chi tiết
- Có thể khái quát được toàn bộ hệ thống

➡ Nhược điểm:

- Tài liệu có thể không đúng vì bị lạc hậu so với thực tế

➡ Sau khi hiểu sơ lược yêu cầu, phân tích viên chuyển chúng thành ‘demo’ cho người sử dụng, và qua quá trình xem xét sửa đổi, bản demo được hoàn chỉnh dần từ tổng quát đến chi tiết – để phân tích viên hiểu rõ chi tiết yêu cầu.

➡ Ưu điểm

- Giúp cho phân tích viên hiểu đúng yêu cầu
- Giúp cho người sử dụng hiểu khả năng của sản phẩm

➡ Nhược điểm

- Khó thống nhất quan điểm sử dụng từ nhiều users
- Khó diễn tả các xử lý tiềm ẩn bên trong hệ thống

- ➡ Pp. truyền thống mang tính chất “cải tiến” hệ thống, dùng CNTT để hỗ trợ mô hình và nghiệp vụ đã có sẵn.
- ➡ Pp. “cải cách”: Thay đổi tiến trình kinh doanh chính để tạo ra sự đột phá bằng cách tái cấu trúc lại các tiến trình kinh doanh (Business Process Reengineering) để tận dụng ưu thế của phương pháp mới hoặc công nghệ mới (E-commerce !).
 - Thay đổi mô hình và nghiệp vụ bằng cách tìm hiểu cách giải quyết vấn đề hiện tại của thế giới.
 - Quan điểm: “Nếu một tổ chức được xây dựng lại từ đầu, thì nó cần phải hoạt động như thế nào?”. Ví dụ: Nhà sách Amazon.com bán sách điện tử thay cho các quyển sách giấy \Rightarrow không có chi phí lưu kho, không có quầy giao dịch và trưng bày, mở rộng kinh doanh, nhưng phải đối mặt với vấn đề “copy rights”.

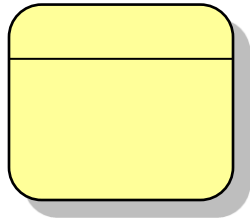
1. Nhận xét và kết luận sơ lược sau khi khảo sát
 - *Workload*: Tần suất, khối lượng cao ở đâu, khi nào.
 - *Effectiveness*: nghẽn cổ chai, xung khắc thông tin, hiệu quả của các báo cáo
 - *Efficiency*: Các công việc tương tự nhau có bị lặp lại ở nhiều nơi không ?
2. Nhận định sơ lược về điểm mạnh, điểm yếu, cơ hội, thách thức (**SWOT**) của hệ thống cũ để định hướng cho việc phân tích, thiết kế hệ thống mới (khắc phục, cải tiến hoặc cải cách)
 1. Nội bộ của tổ chức (**SW**)
 2. Môi trường bên ngoài (**OT**).

- ➔ Sau khi khảo sát và thu thập thông tin mô tả cho hệ thống hiện tại, người phân tích viên cần phải hệ thống hóa lại những gì đã biết về:
 1. kiểm tra phát hiện thiếu sót hoặc mâu thuẫn trong cách hiểu biết của mình
 2. Chia sẻ hiểu biết của mình với nhóm công tác
- ➔ Việc này chỉ thực sự hiệu quả khi *các đặc trưng quan trọng nhất của hệ thống được làm nổi bật* (sáng tỏ) dễ hiểu, các chi tiết không quan trọng đã được loại bỏ.
- ➔ Ngôn ngữ tự nhiên thường gây hiểu lầm, và không trợ giúp cho việc khái quát hóa nên người ta thay thế chúng bằng các mô hình (models).

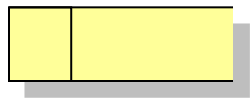
- ➡ Mô hình là phương tiện để diễn tả các đặt trưng quan trọng nhất của hệ thống theo một quan điểm phân tích nào đó.
- ➡ Trong hệ thống thông tin, mô hình là một hệ thống lược đồ sử dụng các ký hiệu, hình ảnh gợi nhớ để diễn tả ý như DFD, ERD, UML.
- ➡ Mô hình có 3 đặc tính cơ bản:
 1. *Ngữ pháp* (notations): là các quy tắc sử dụng các ký hiệu hình thức cho mô hình, để loại bỏ những mô tả vô lý hoặc tối nghĩa.
 2. *Ngữ nghĩa* (semantics): là nội dung (ý) cần diễn tả lại.
 3. *Ngữ cảnh* (context): là kiến thức chung giữa người xem và người tạo ra mô hình để nội dung ngữ nghĩa của mô hình được truyền đạt trọn vẹn cho người đọc. Vì lý do này, một lược đồ cho hệ thống chỉ được tạo ra chỉ từ một quan điểm phân tích nào đó.

Lược đồ dòng dữ liệu - Data Flow Diagram 17

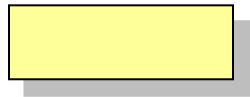
➔ DFD là lược đồ sử dụng 4 ký hiệu cùng với các quy tắc vẽ để diễn tả các dòng dữ liệu di chuyển trong hệ thống.



Process : Là một hành động hoặc một hệ thống con xử lý trên dữ liệu (biến đổi, lưu trữ hoặc phân phối dữ liệu).



Data store : Là bộ phận dùng để lưu trữ dữ liệu, như tập tin, hồ sơ, CSDL,...



Source/Sink : Là thành phần phát sinh dữ liệu (source) cho hệ thống, hoặc tiêu thụ dữ liệu (sink) từ hệ thống.



Data flow : dòng dữ liệu cơ bản của lược đồ, chỉ ra 1 nội dung dữ liệu (không đổi) được gửi từ đâu và đi đến đâu.

➔ Quy ước

- Dùng **Động từ** để đặt tên cho Process
- Dùng **Danh từ** để đặt tên Data store, Source, Sink và Data flow

- ➔ Dòng dữ liệu được tạo thành từ các vật mang dữ liệu (ví dụ: chứng từ, hóa đơn, phiếu nhập xuất kho,...) qua các xử lý trung gian biến đổi dữ liệu theo các quy tắc quản lý của tổ chức (business rules).
- ➔ Các dòng vật chất, tiền tệ, dịch vụ, thông tin quan trọng trong hệ thống là nguồn gốc làm phát sinh dòng dữ liệu trên lược đồ.

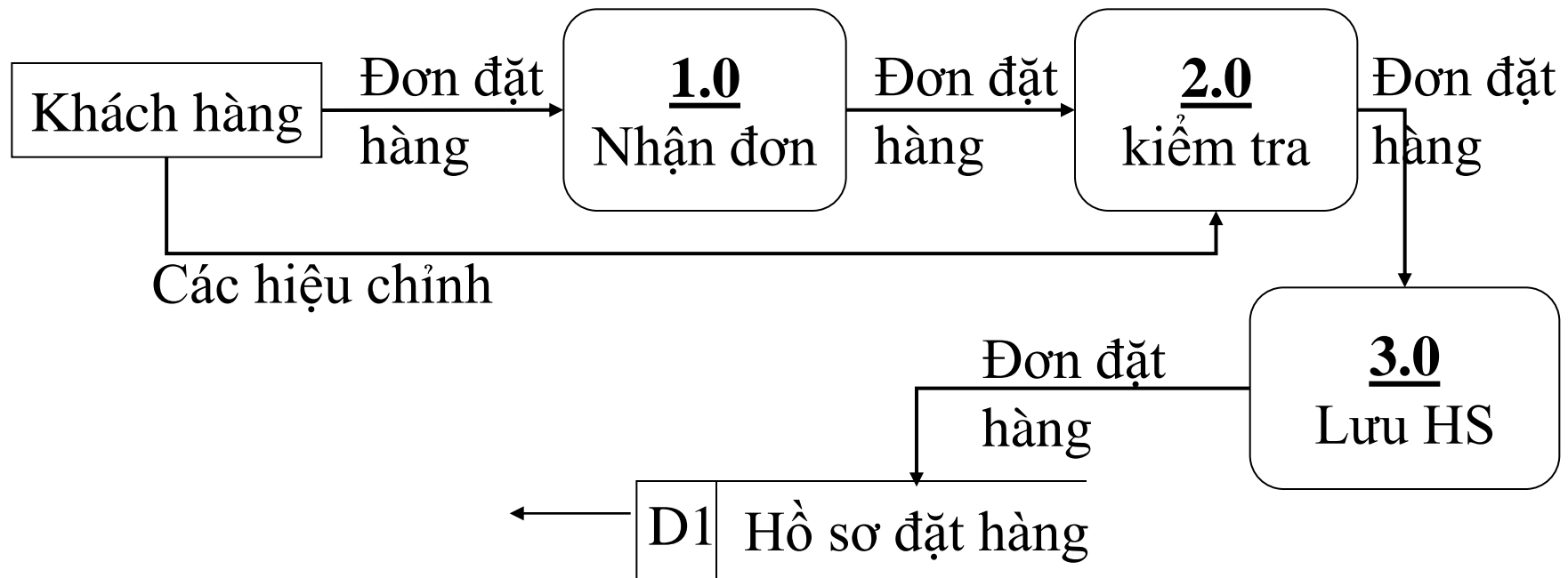
Dữ liệu trên dòng dữ liệu

Tên	Tuổi	hrs/week
Smith	25	44
Chen	42	35

Dòng dữ liệu của tiến trình
tính lương trong tổ chức

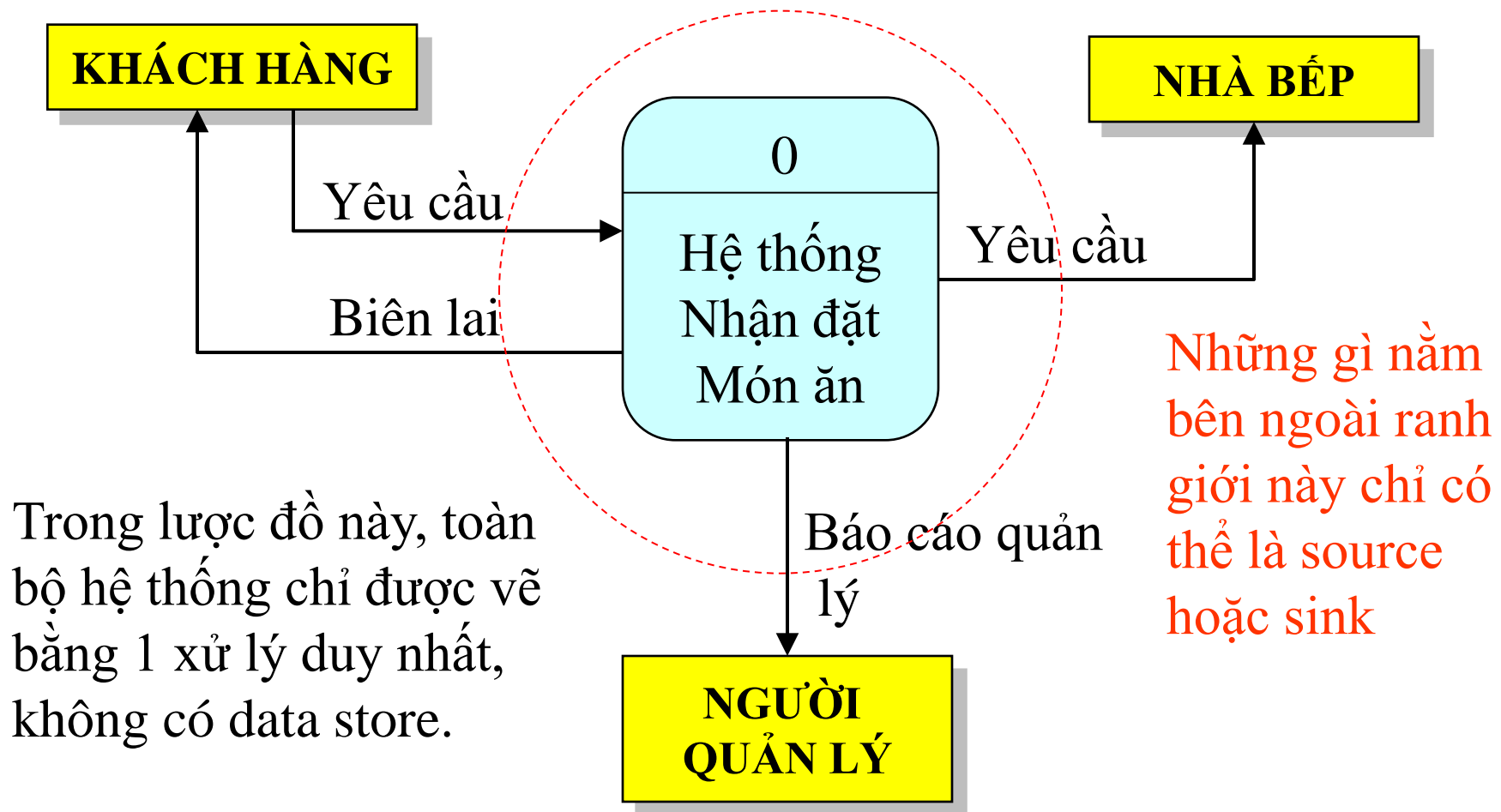


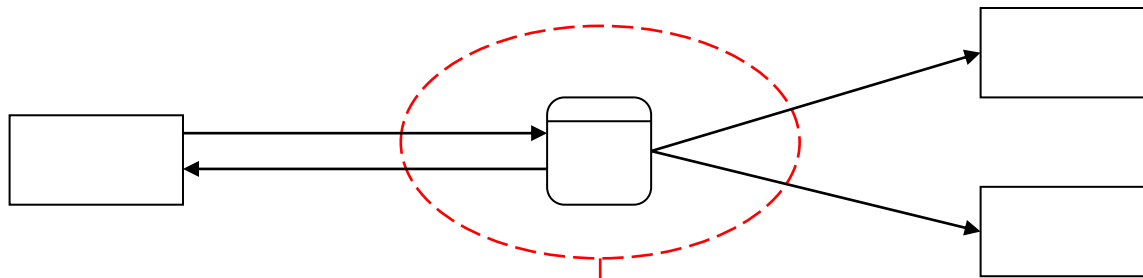
- ➔ Các dòng dữ liệu trong hệ thống có kiểm soát như công ty, doanh nghiệp đều phải tuân thủ **các quy tắc quản lý (business rules)**; vì vậy các quy tắc quản lý (và quy trình) của hệ thống là cơ sở để thiết lập các dòng dữ liệu trong lược đồ DFD.
- ➔ Ví dụ: Bộ phận giao dịch nhận đơn đặt hàng từ khách hàng, kiểm tra đơn đặt hàng để hiệu chỉnh nếu cần, sau đó lưu vào hồ sơ đặt hàng.



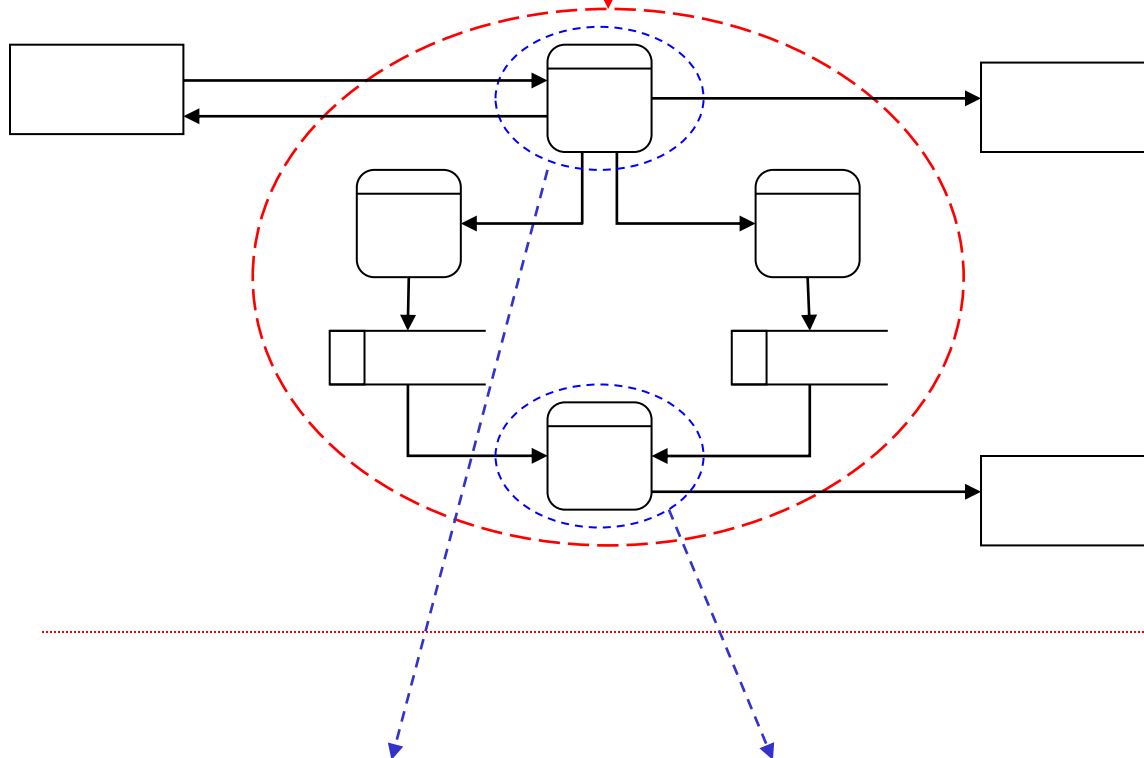
- ➔ Lược đồ ngữ cảnh (context diagram) là lược đồ DFD được dùng để mô tả khái quát toàn bộ các dòng dữ liệu đi từ nguồn (SOURCE) vào hệ thống và đi từ hệ thống ra đến đích (SINK) mà không quan tâm đến các xử lý biến đổi và lưu trữ dữ liệu bên trong hệ thống.
- ➔ Lược đồ ngữ cảnh cho biết hệ thống cần chuyển giao (dữ liệu) gì, và nó nhận được (dữ liệu) gì từ môi trường (các SOURCE/SINK) bên ngoài hệ thống.
- ➔ Ví dụ: nhà hàng Hoosie Burger có một hệ thống “đặt hàng các món ăn” được mô tả tổng quát theo các quy tắc quản lý như sau: hệ thống sẽ nhận yêu cầu từ khách hàng, in biên lai thanh toán tiền cho khách hàng, chuyển yêu cầu cho nhà bếp thực hiện và in các báo cáo quản lý cho người quản lý nhà hàng vào cuối ngày.

Ngữ cảnh của hệ thống nhận đặt món ăn của Hoosie Burger





Ngữ cảnh /
DFD mức n



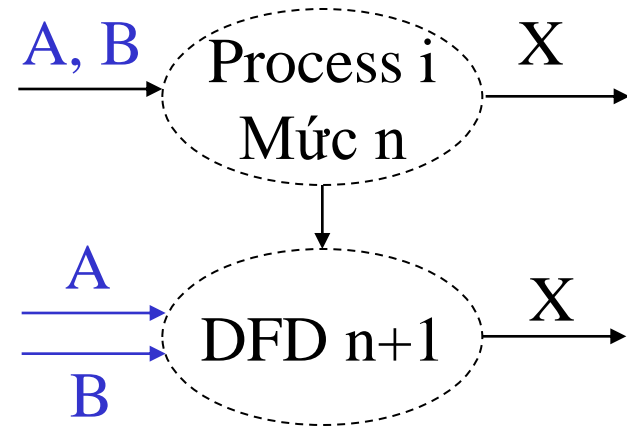
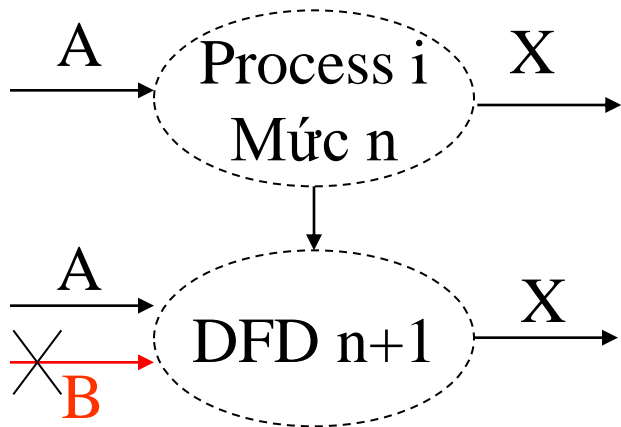
DFD mức n+1

Một lược đồ DFD
mức n+1 là lược đồ
DFD mô tả chi tiết
hơn cho một xử lý
trong lược đồ DFD
mức n

DFD mức n+2

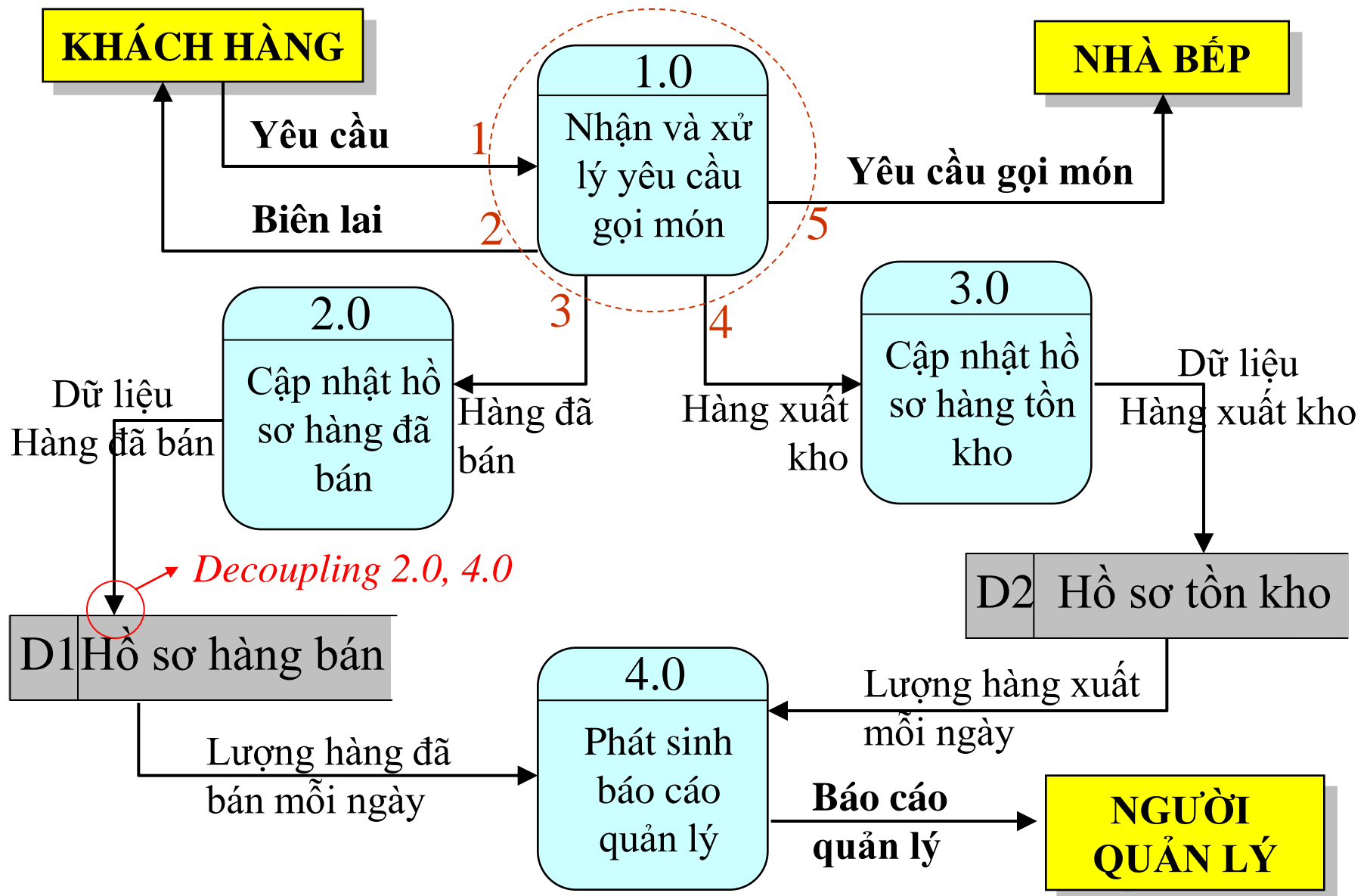
1. Nếu một đối tượng chỉ có outputs, chắc chắn đối tượng đó phải là source. Nếu một đối tượng chỉ có inputs, nó phải là sink.
2. Một xử lý phải có cả inputs lẫn outputs. Không có xử lý nào chỉ có inputs mà không có outputs, hoặc ngược lại. Tương tự, Datastore phải có bộ dữ liệu đi vào (D_{in}) và đi ra (D_{out}), và $D_{in} \supseteq D_{out}$.
3. Một dataflow là một mũi tên phải có nhãn và có duy nhất một hướng để chỉ rõ nơi đi và nơi đến của dữ liệu. Do đó, nếu một nội dung dữ liệu được chuyển đi và nhận về giữa hai đối tượng thì nó phải được vẽ bằng 2 mũi tên (theo 2 hướng ngược nhau).
4. Không có dòng dữ liệu trực tiếp giữa các data store, source, sink; vì đây là những đối tượng “thụ động”; để di chuyển dữ liệu giữa các đối tượng này cần phải có ít nhất một xử lý của hệ thống.
5. Không có dòng dữ liệu rẽ nhánh (hoặc gộp) có nội dung (nhãn) khác nhau. Nội dung dữ liệu ở các nhánh phải giống y như nhau.
6. Không có dòng dữ liệu trực tiếp đi từ một xử lý đến chính nó (vì một xử lý không cần gửi dữ liệu cho chính nó).

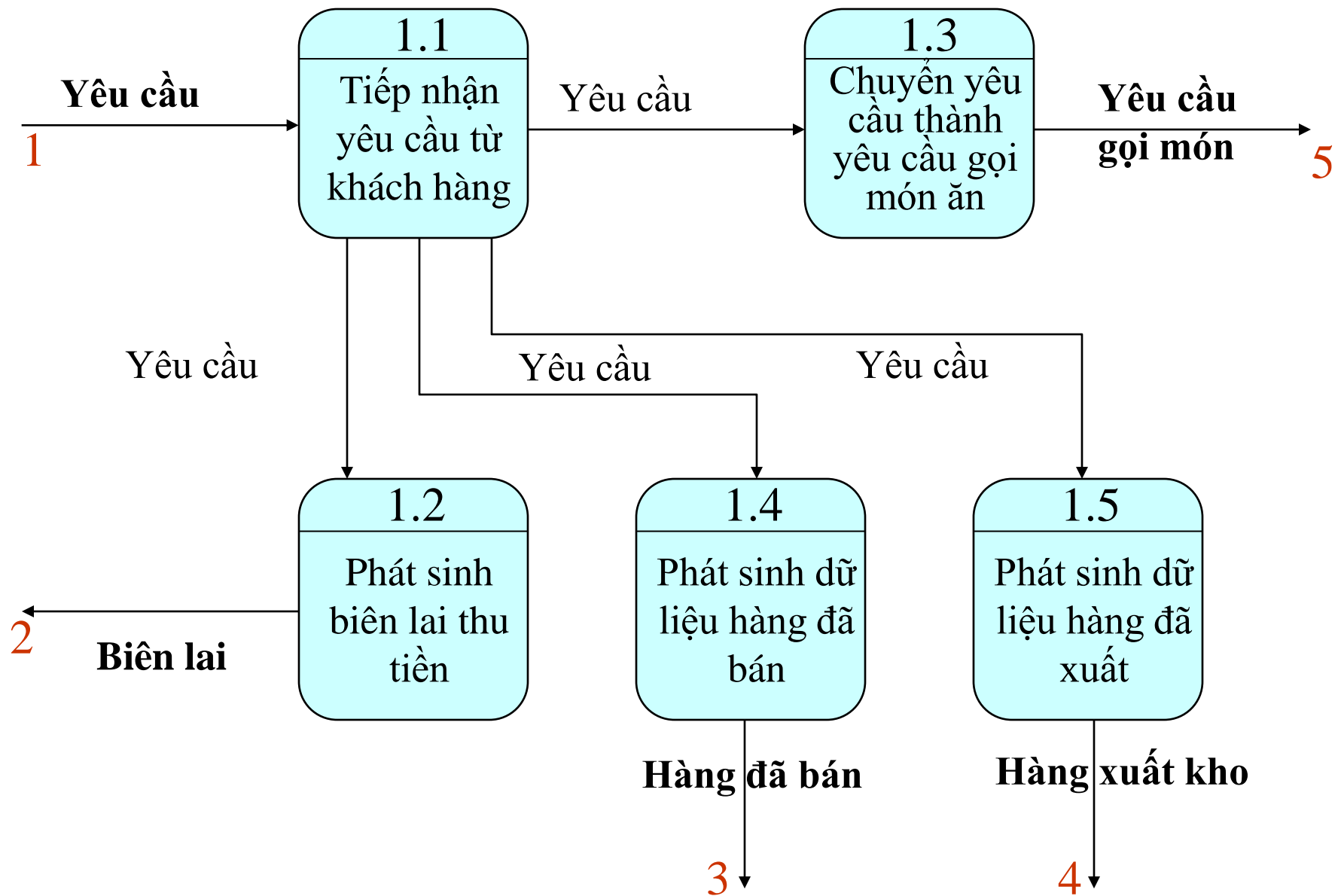
1. Nếu một xử lý i (mức n) được phân rã thành một lược đồ DFD mức $n+1$ cho xử lý này thì nội dung dữ liệu vào ra của xử lý i phải được bảo toàn (không thêm/bớt).
2. Nếu dòng dữ liệu ở mức n là dạng tổng quát (như “thông tin khách hàng”,...) trong khi các dòng dữ liệu ở lược đồ DFD mức $n+1$ là dạng chi tiết (“tên khách”, “địa chỉ”,...) thì phải sử dụng từ điển dữ liệu để liên kết chúng với nhau.

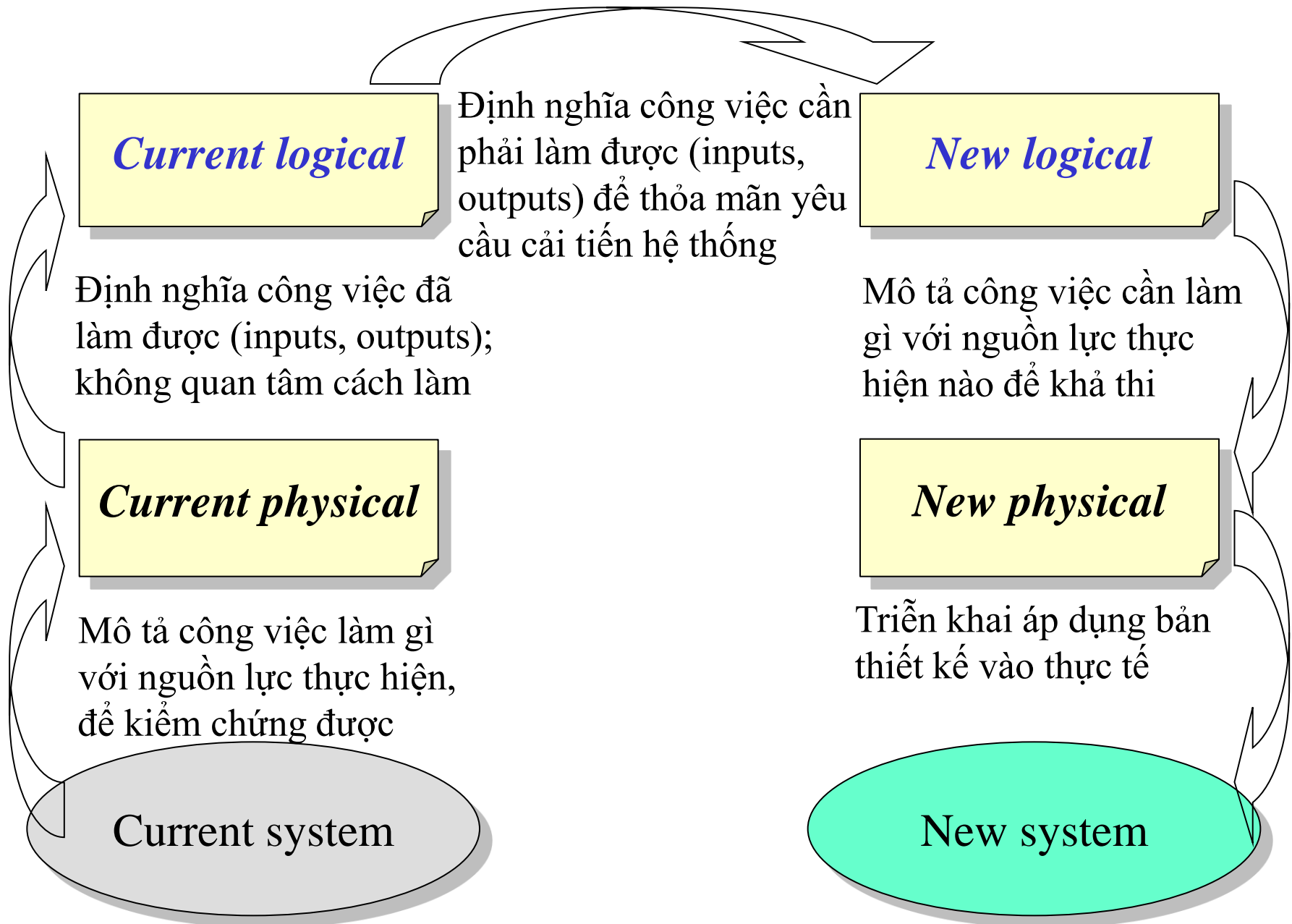


Hệ thống đặt món ăn của nhà hàng Hosier Burger được mô tả như sau:

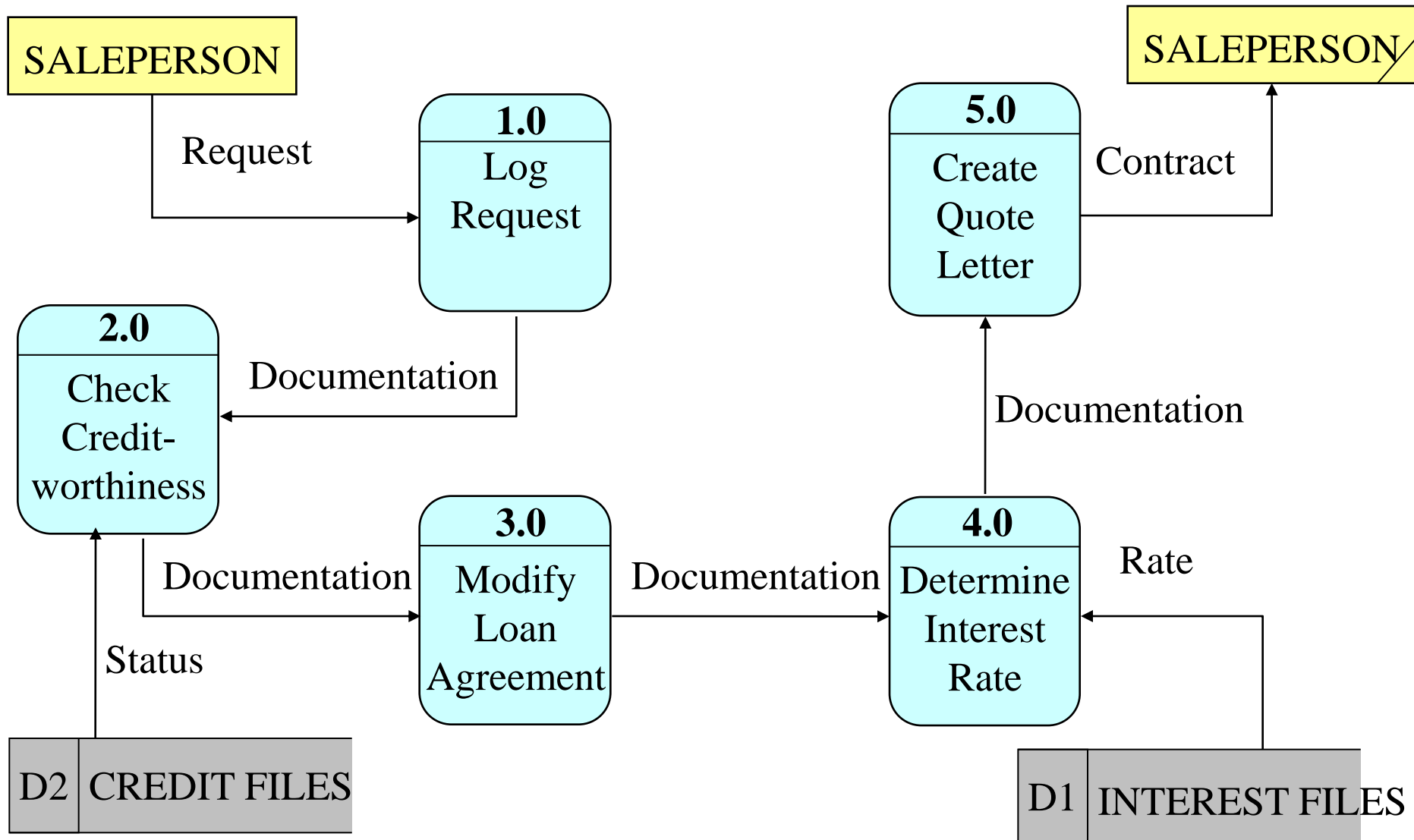
1. Chức năng “tiếp nhận và xử lý yêu cầu gọi món”: nhận yêu cầu từ khách hàng, chuyển yêu cầu gọi món đến nhà bếp, phát sinh biên lai thu tiền cho khách, tạo ra dữ liệu ‘hàng đã bán’ và ‘hàng xuất kho’ cho 2 chức năng ‘cập nhật hồ sơ hàng bán’ và ‘cập nhật hồ sơ hàng tồn kho’.
2. Cập nhật hồ sơ hàng đã bán: cập nhật hồ sơ hàng đã bán theo đúng khuôn mẫu dữ liệu của hồ sơ này.
3. Cập nhật hồ sơ hàng tồn kho: Cập nhật hồ sơ hàng tồn kho theo đúng khuôn mẫu của hồ sơ này.
4. Phát sinh báo cáo quản lý: lấy dữ liệu hàng đã bán mỗi ngày từ hồ sơ hàng đã bán và dữ liệu hàng xuất kho mỗi ngày từ hồ sơ hàng tồn kho để in báo cáo cho người quản lý nhà hàng.



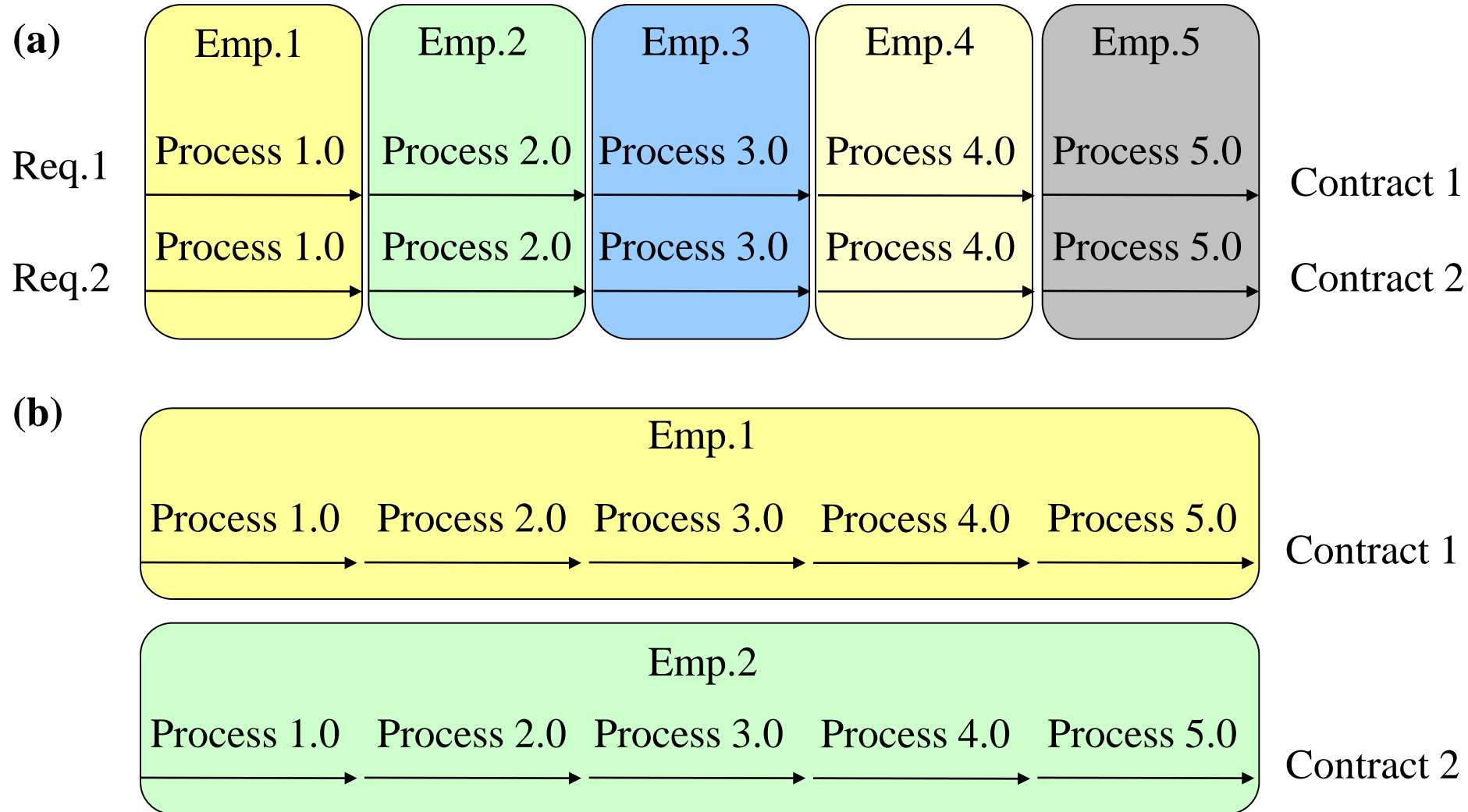




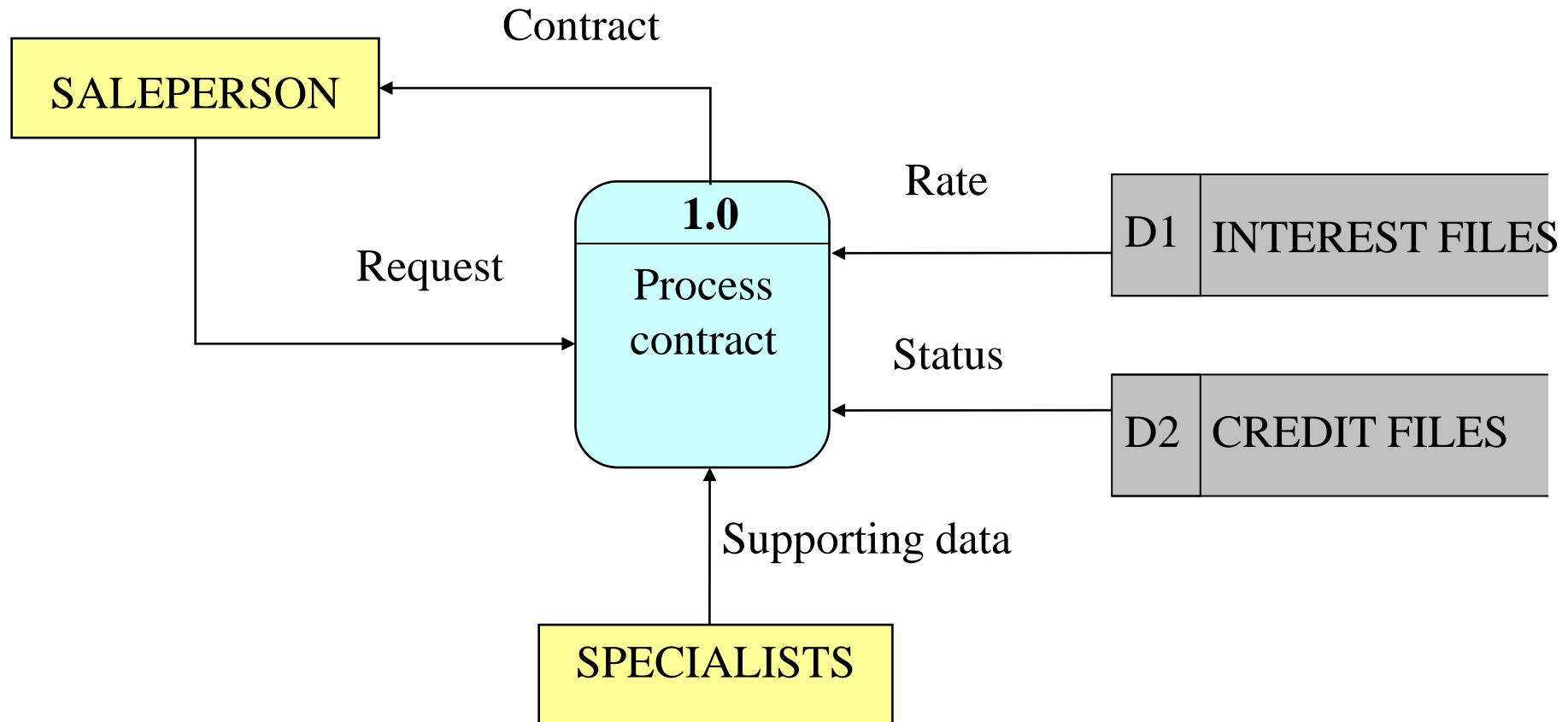
Hiện trạng: Current Logical DFD-mức 0



Phân tích dòng dữ liệu

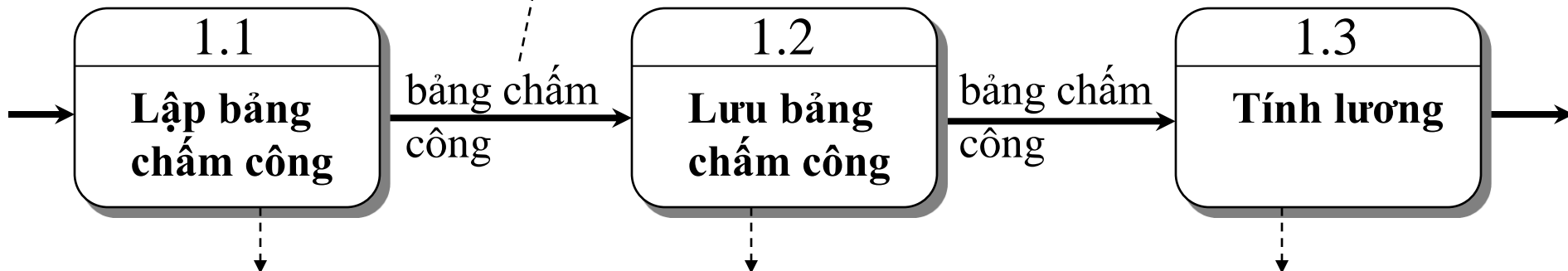


Kết quả: New Logical DFD mức 0



- ➔ DFD trợ giúp phân rã hệ thống, nhưng tên của các xử lý không đủ mô tả các xử lý. Processing Logic trợ giúp mô tả rõ hơn cấu trúc xử lý bên trong Process dựa trên các Business Rules (quy tắc quản lý) của tổ chức.
1. Structured English: mô tả ngắn gọn về các quy tắc xử lý bằng ngôn ngữ tự nhiên (tiếng Việt hoặc tiếng Anh)
 2. Decision Table và Decision Tree: thể hiện các quy tắc xử lý “IF <điều kiện> THEN <xử lý>” bằng bảng hoặc lưu đồ rẽ nhánh (tree) trong trường hợp có nhiều quy tắc xử lý phức tạp
 3. Data Dictionary: là từ điển dữ liệu giải thích ý nghĩa của từng thành tố dữ liệu được các processes sử dụng hoặc tạo ra.

Tên	LoạiHĐ	Tuổi	hrs/week
Smith	H	25	44
Chen	S	42	35



Process logic 1.1

Nơi: **Tổ sản xuất**

Xử lý: **Cuối tuần**

Ghi số giờ làm của mỗi người thành bảng chấm công (BCC) nộp P.Tổ chức

Process logic 1.2

Nơi: **P.Tổ chức**

Xử lý: **Khi nhận BCC**

Tập hợp, vào sổ tất cả các bảng chấm công, chuyển cho P.Kế toán.

Process logic 1.3

Nơi: **P.Kế toán**

Xử lý: **Khi đủ BCC**

IF Tg > 40 **THEN**

Trả = 40 * mức

Trả + = (Tg-40)*mức*1.2

ELSE Trả = Tg * mức

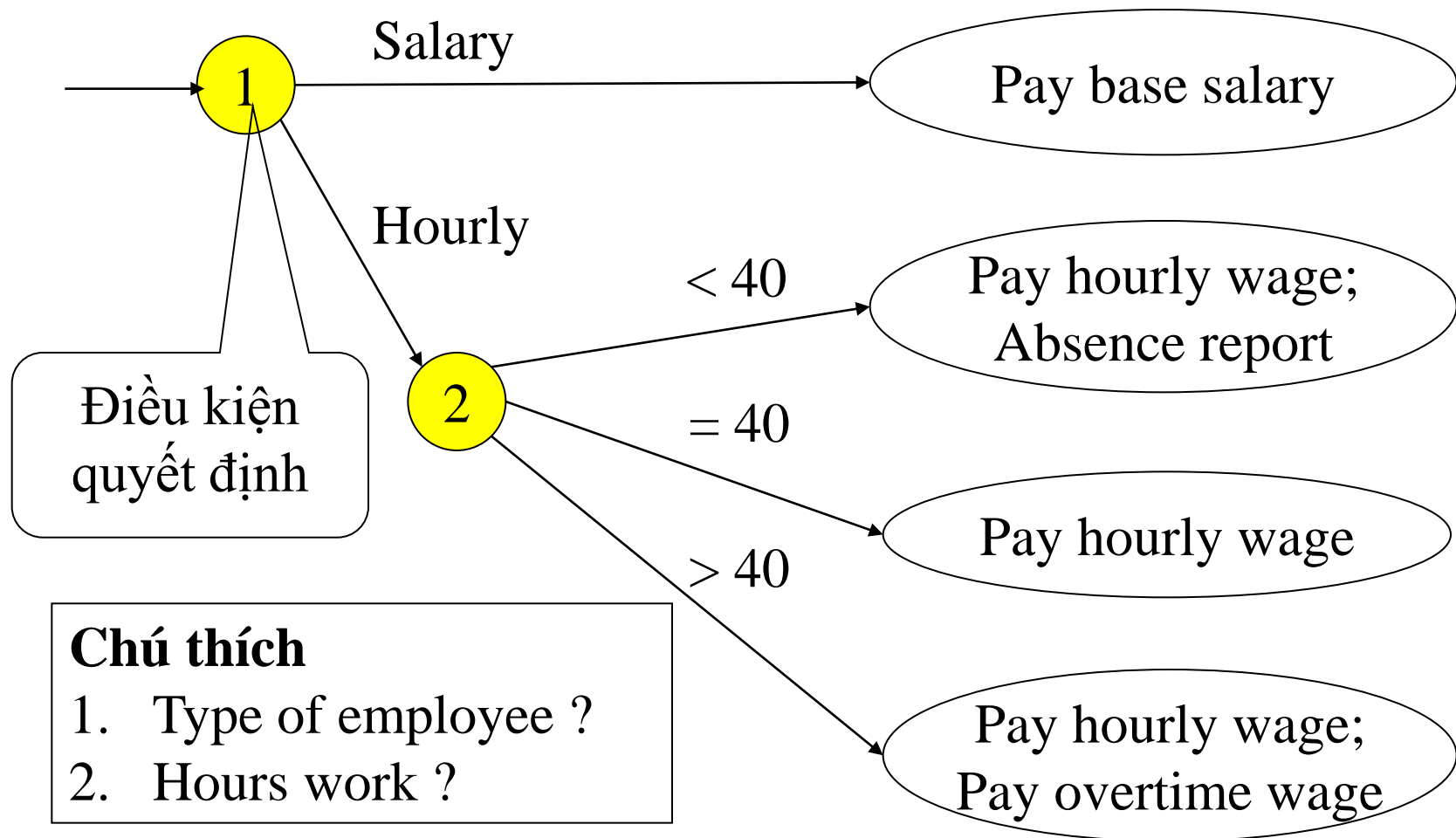
➔ Decision Table thay thế Structured English nhằm thể hiện các luận lý phức tạp.

ĐIỀU KIỆN	QUY TẮC ÁP DỤNG			
(1) Employee type	S	H	H	H
(2) Hours worked	-	<40	40	>40
CÁC XỬ LÝ				
Pay base salary	✓			
Calculate hourly wage		✓	✓	✓
Calculate Over time				✓
Produce absence report		✓		

S: Salary

H: Hours

⇒ Decision tree minh họa cấu trúc chọn hành động xử lý theo điều kiện bằng lưu đồ rẽ nhánh.



- ➡ Là một mô tả cho từng phần tử dữ liệu được dùng trong hệ thống.
Thông thường, từ điển dữ liệu được lập thành một bảng gồm các cột như ví dụ sau:

Tên gọi	Ý nghĩa	Cấu trúc dữ liệu	Nguồn gốc
Bảng chấm công	Ghi số giờ công của từng nhân viên	Tên, tuổi, số giờ công trong tuần	Process 1.1
Tiền lương	Tiền công theo giờ công trong tuần	Mức chuẩn + phụ trội	Process 1.3
Mức chuẩn	Mức lương trả trong định mức 40 giờ / tuần	Giờ công trong định mức * giá định mức	Process 1.3
Phụ trội	Mức trả vượt định mức 40 giờ / tuần	Giờ công vượt trội * giá định mức * 1.5	Process 1.3

➡ Công ty Wonder Widget (WWC) có một hệ thống giám sát các *đơn đặt hàng* từ khách hàng. WWC lấy *mã số của các món hàng* được yêu cầu từ danh mục hàng (Master Item Number List), lấy *đơn giá* của món hàng từ hồ sơ giá (Master Price List), và kiểm tra *mức tín dụng* của khách hàng trong hồ sơ tín dụng (Credit Rating File). Các *phiếu đặt hàng được chấp nhận* sẽ lưu trong hồ sơ đặt hàng được chấp nhận (Accepted Orders File), các *phiếu đặt hàng bị từ chối* được lưu trong hồ sơ đặt hàng bị từ chối (Rejected Orders File). Hệ thống kho được kiểm tra để xem *số lượng hàng tồn* có đủ cho các phiếu đặt hàng được chấp nhận trong hồ sơ đặt hàng được chấp nhận hay không. Nếu đủ, *phiếu đặt hàng được phê duyệt* và gửi đến trung tâm giao hàng. Nếu không đủ, phiếu sẽ bị từ chối và đưa vào hồ sơ đặt hàng bị từ chối. Hồ sơ đặt hàng bị từ chối được dùng để phát sinh *báo cáo phiếu đặt hàng bị từ chối* cho giám đốc của WWC. Hãy vẽ lược đồ ngữ cảnh và DFD level 0 cho hệ thống.

➡ Các từ in nghiêng là nội dung dữ liệu, các từ gạch dưới là source hoặc sink. Hệ thống không giám sát việc giao hàng và thanh toán tiền.

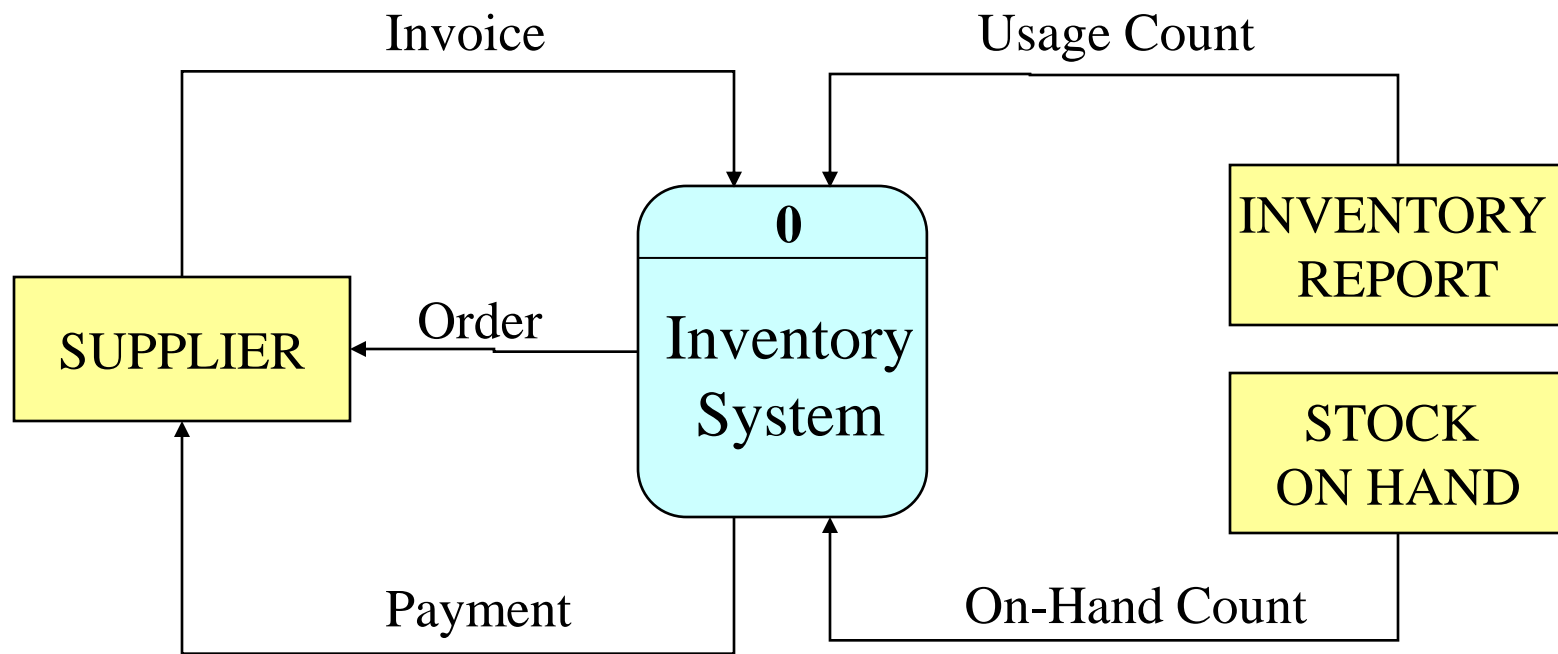
- ➡ DFD và Processing Logic chỉ ra *làm thế nào, ở đâu và khi nào* dữ liệu được xử lý, nhưng không chỉ ra *định nghĩa, cấu trúc và các quan hệ* của dữ liệu.
- ➡ Entity Relationship Diagram (ERD) là lược đồ thể hiện cấu trúc trừu tượng hóa của dữ liệu trong tổ chức, dựa trên khái niệm thực thể (entity) và quan hệ (relationship) giữa các thực thể, để nhằm thể hiện nội dung, ý nghĩa của dữ liệu trong hệ thống.

1. *Định nghĩa các thực thể*, dựa trên vai trò, ý nghĩa của thực thể đối với hệ thống. Nên chọn danh từ để dùng làm tên cho thực thể, vd: MONHOC, SINHVIEN, KHOA,...
2. *Định nghĩa các quan hệ giữa các thực thể*. Tên của các quan hệ thường được diễn tả bằng động từ để chỉ các hành động, sự kiện liên kết các thể hiện trong các thực thể có quan hệ nhau.
3. *Xác định các thuộc tính của thực thể và quan hệ*. Thuộc tính của thực thể (hoặc quan hệ) là những đặc tính mà tất cả các thể hiện của thực thể (hoặc quan hệ) đều có. Thêm thuộc tính để tăng tính mô tả, hoặc để có thể dữ liệu phân biệt các thể hiện. Bỏ bớt thuộc tính nếu chúng dư thừa hoặc không liên quan đến vai trò, ý nghĩa của thực thể trong hệ thống.
4. *Xác định cardinality cho mỗi quan hệ*.

1. Mỗi nhân viên thuộc 1 phòng, một phòng có ít nhất 1 hay nhiều nhân viên. Không có nhân viên nào không thuộc phòng nào. Mỗi nhân viên có mã số, tên và mức lương. Mỗi phòng có số phòng, tên và nhiệm vụ.
2. Mỗi giảng viên (có mã số, tên, chuyên môn) dạy nhiều môn học (có mã mh, tên, nội dung, số tiết) trong các phòng học khác nhau (phân biệt bằng số phòng).
3. Mỗi sinh viên sau mỗi lần thi một môn học sẽ có một điểm xác định cho lần thi đó.
4. Các quầy hàng có mã số và vị trí, bán nhiều mặt hàng có mã số và tên mặt hàng; mỗi mặt hàng còn có một giá niêm yết tại quầy hàng đó.
5. Mỗi bệnh nhân nội trú có mã số và tên được bệnh viện cấp cho một giường ngủ có mã số giường đặt trong phòng bệnh có mã số phòng và tên phòng. Khi nhận và trả giường thì ngày nhận và ngày trả cũng được bệnh viện ghi vào hồ sơ quản lý.

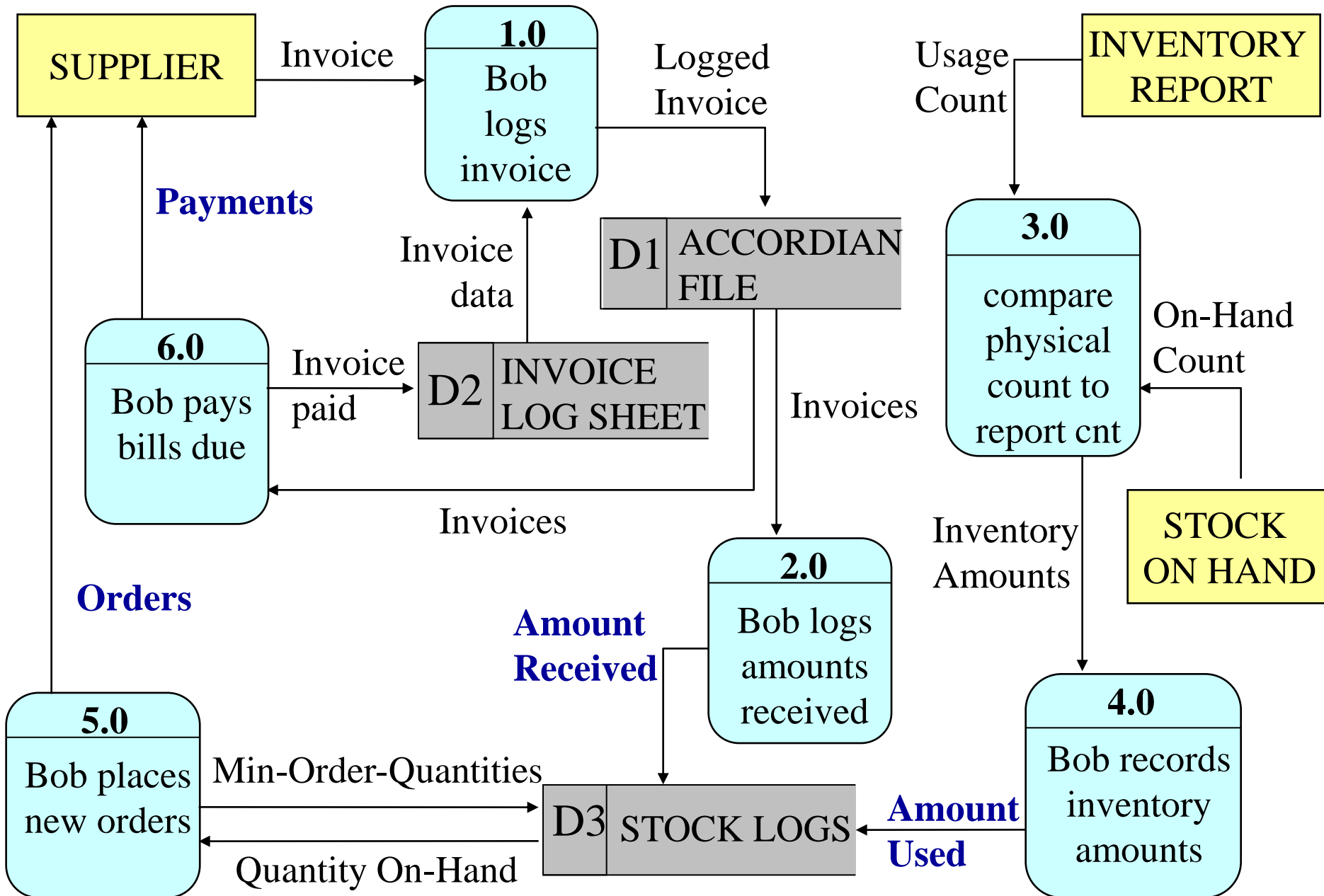
- Hàng hóa được *xuất cảng* trong các container và mỗi container được định danh bởi số của container, đích đến và kích cỡ của nó. Đại lý vận chuyển (Shipping Agent) chịu trách nhiệm *gom nhóm* các container vào một lô hàng và cho lô hàng một số định danh và giá trị của lô. Các con tàu, có số của con tàu, tải trọng và quốc gia đăng ký, *thực hiện* các chuyến hải trình, mỗi chuyến hải trình *mang* một số lô hàng đi đến đích của chúng. Mỗi chuyến hải trình được cho số hải trình. Hãy vẽ lược đồ ERD chỉ ra tất cả các thực thể, thuộc tính, và quan hệ.
- Để vẽ các ERD, chúng ta cần xác định các thực thể trước, sau đó là quan hệ giữa các thực thể, và cuối cùng là các thuộc tính cho thực thể và quan hệ. Các từ gạch dưới là các thực thể, và các động từ in nghiêng là các quan hệ giữa các thực thể. Trong mô tả không nêu rõ cardinality (số quan hệ), do đó số quan hệ có thể được cho bằng cách suy diễn hợp lý.

Ví dụ lược đồ ngữ cảnh của hệ thống quản lý kho nhà hàng Hoosie Burger



1. Current physical: Level 0 DFD

43

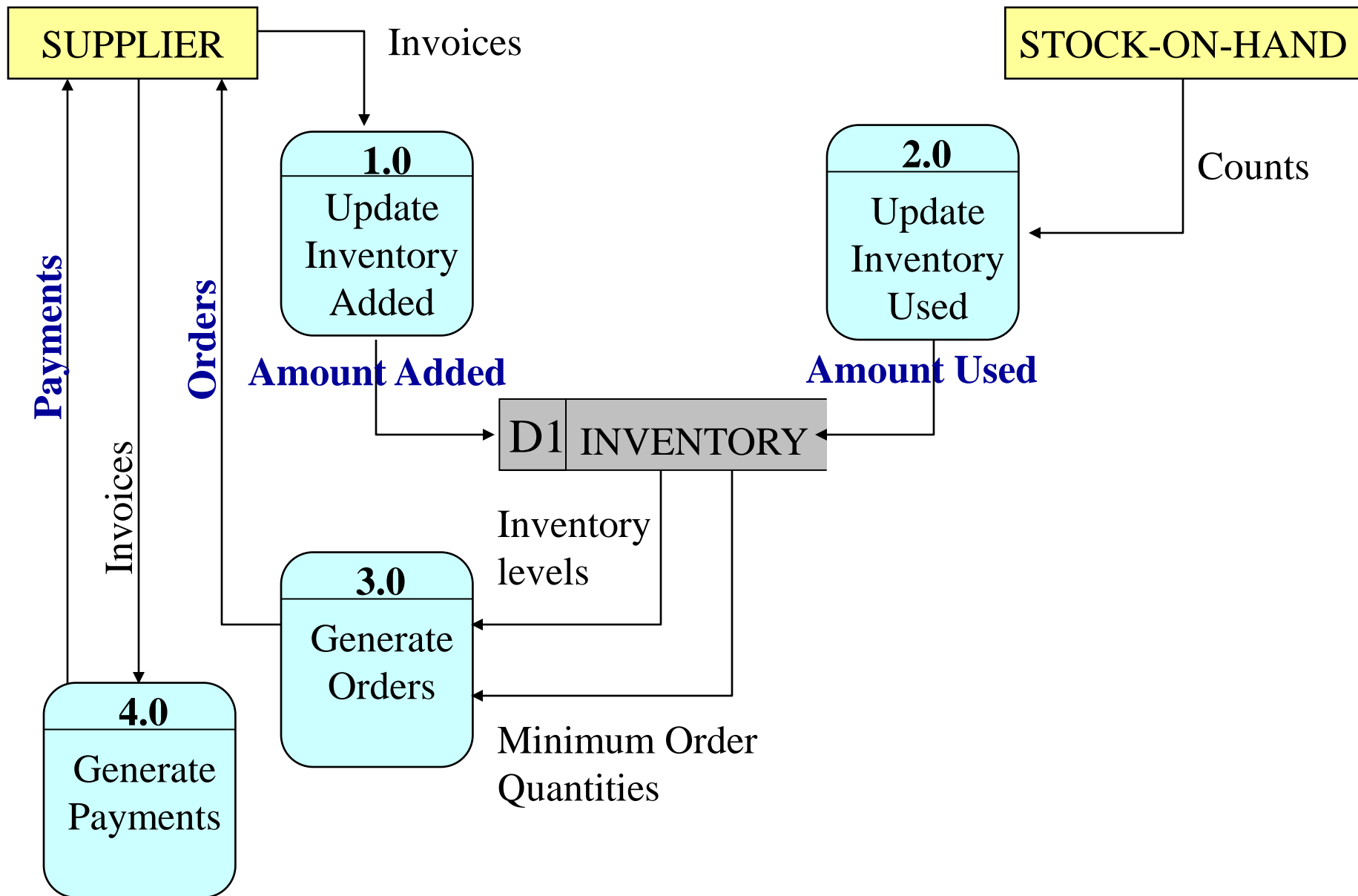


- ➡ Trong ví dụ hệ thống quản lý kho của nhà hàng Hoosie Burger, lược đồ DFD vật lý chỉ ra rằng 2 dòng dữ liệu từ kho đến process 3.0 (Usage count và On-hand count) đều từ một nguồn là kho => chỉ nên dùng một dòng dữ liệu.
- ➡ Hệ thống cần quản lý trực tiếp nguyên liệu đưa vào kho bằng hóa đơn từ nhà cung cấp (invoice) và giám sát mức độ tồn kho bằng cách đếm (on-hand count) để quản lý.
- ➡ Như vậy Process 3.0 & dòng dữ liệu Usage count trong DFD-0 vật lý không còn cần thiết.
- ➡ Data store D2 lưu số liệu hóa đơn để theo dõi thanh toán tiền hàng (không có dòng dữ liệu đi ra) và Data store D1 chỉ để lưu tạm số lượng hàng nhập kho cho Data store D3 nên D1, D2 không cần thiết.

- ⇒ **Các xử lý chính:** để phục vụ cho mục đích “Vật tư/hàng hóa trong kho phải đủ để sử dụng” gồm:
1. Kiểm soát tất cả những thứ đưa vào kho (1.0, 2.0) => Các xử lý 1.0 & 2.0 gọi chung là xử lý “Update Inventory Added”
 2. Kiểm soát tất cả những thứ lấy ra khỏi kho (3.0, 4.0) => xử lý 3.0 và 4.0 gọi chung là xử lý “Update Inventory Used”
 3. Đặt mua thêm, nếu thiếu (5.0) => “Generate orders”
 4. Trả tiền (6.0) => “Generate payments”
- ⇒ **Dữ liệu chính:** Gồm **Amount received**, **Amount used**, **Orders**, và **Payments**

3. Current Logical: Level 0 DFD

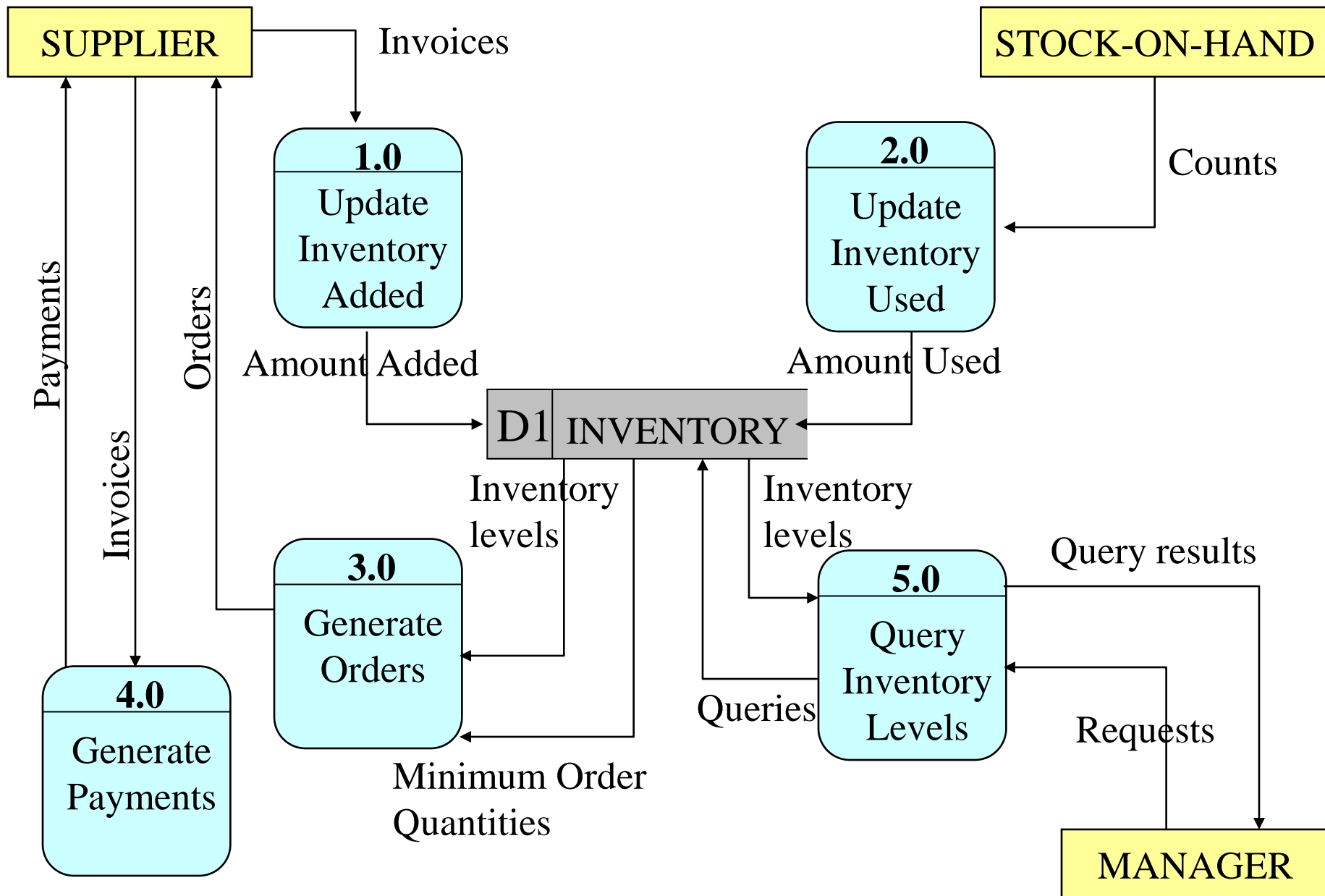
46



- ➡ Qua khảo sát, User (Bob) mong muốn rằng:
 1. Dữ liệu chuyển hàng mới về cần phải đưa vào hệ thống xử lý tự động (không dùng log sheet).
 2. Tự động phát sinh đơn đặt hàng.
 3. Truy vấn số lượng kho bất kỳ lúc nào.
- ➡ Mục đích của New Logical DFD là chỉ ra các chức năng nào cần làm, và các dòng dữ liệu nào cần sử dụng.
- ➡ Yêu cầu 1 & 2 không cần thay đổi vì DFD đã được trừu tượng hóa - không phụ thuộc vào cách thực hiện thủ công hay tự động.
- ➡ Yêu cầu 3 (process 5.0) cần được thêm vào lược đồ hiện hữu (Current Logical DFD) để có lược đồ mới: New Logical DFD

4. New logical : Level 0 DFD

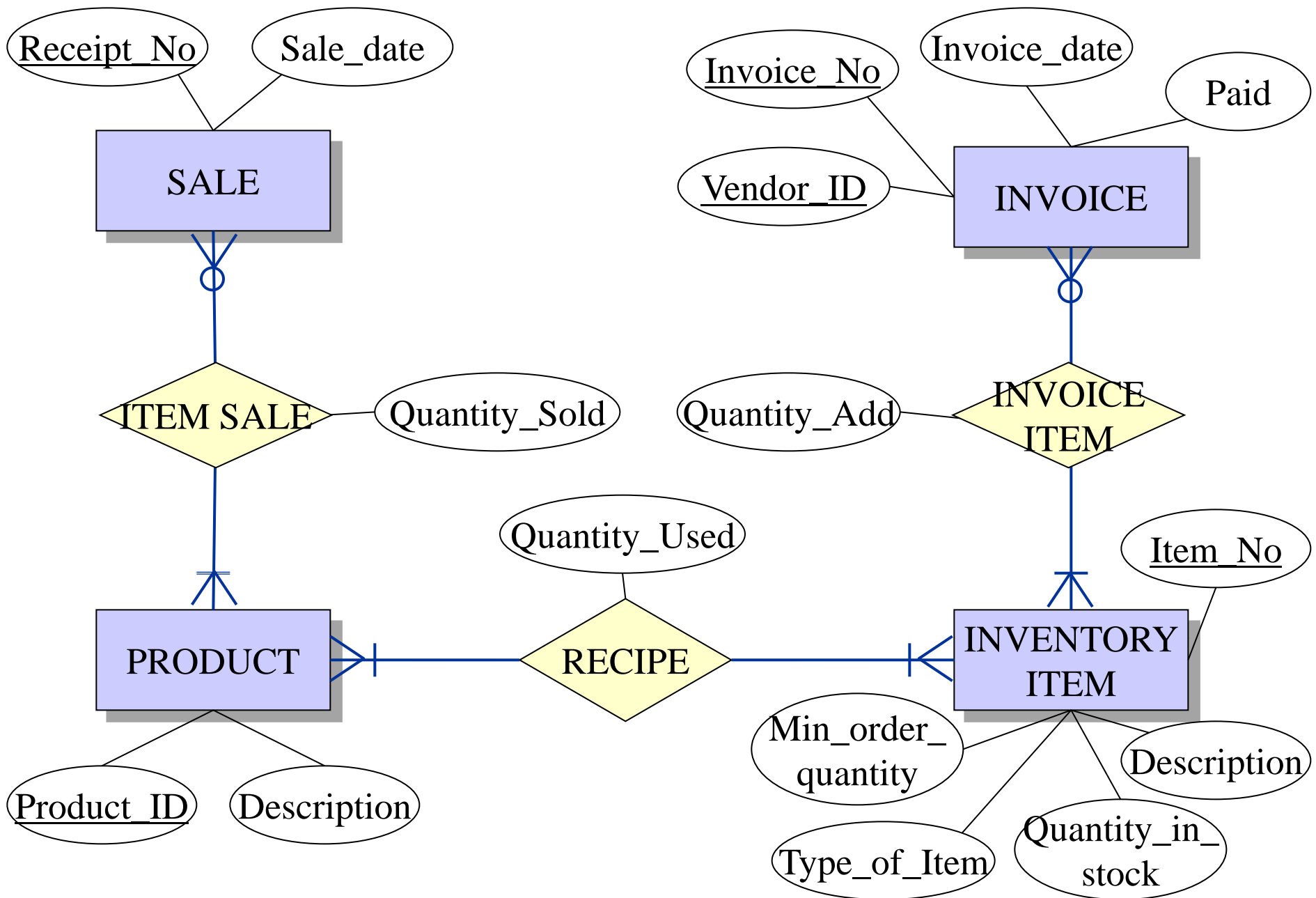
48

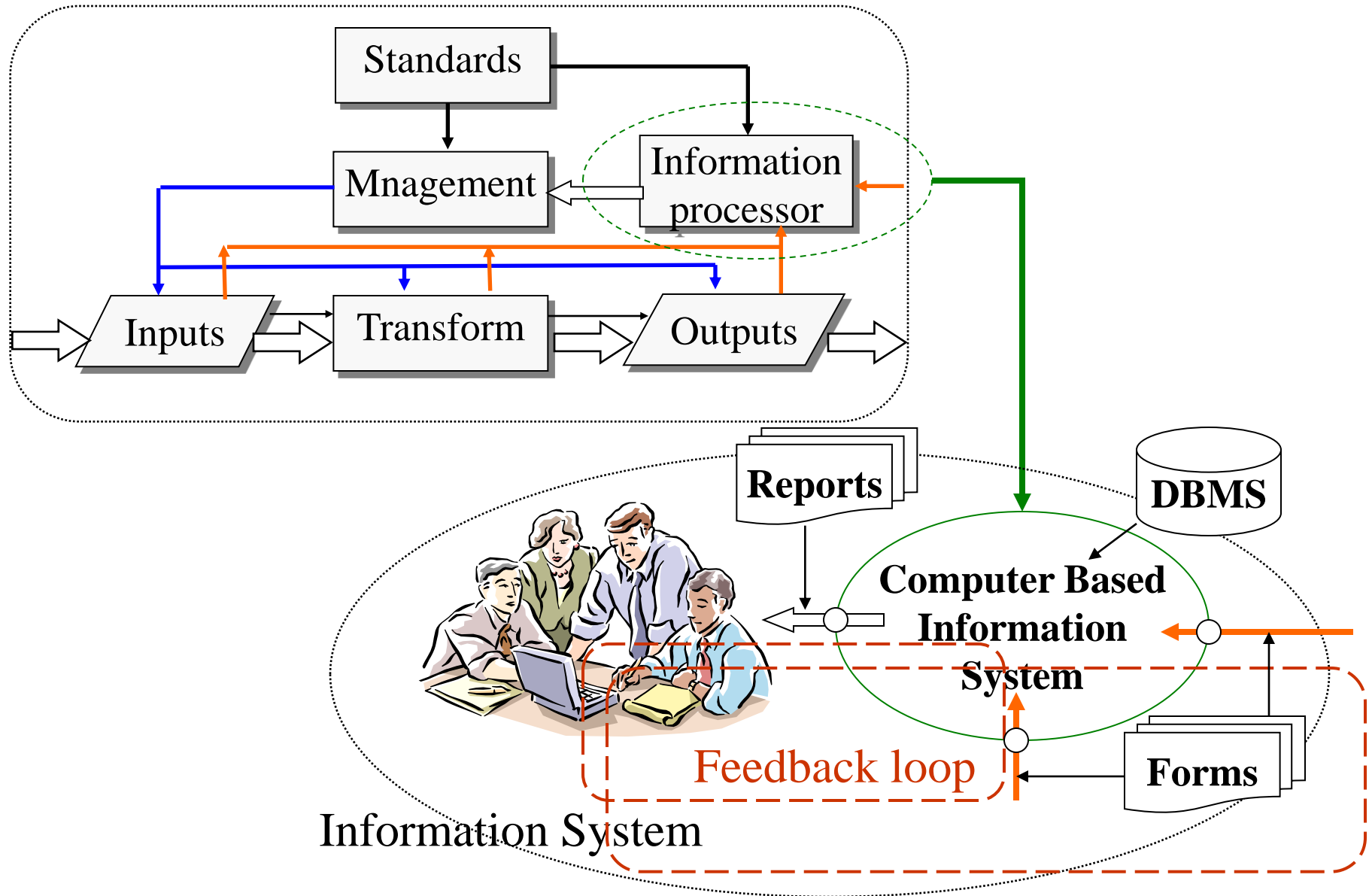


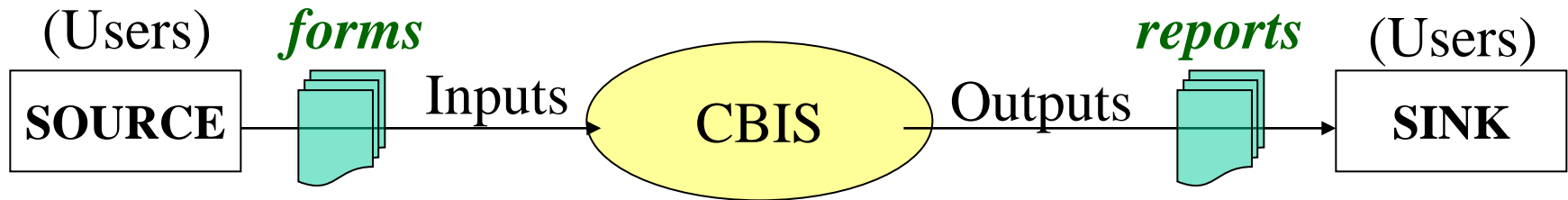
- ➔ Các thay đổi trong kho phát sinh bởi 2 sự kiện: Khi nhận được hàng từ nhà cung cấp (và hóa đơn kèm theo từ nhà cung cấp), và tiêu thụ nguyên liệu khi làm thức ăn bán cho khách hàng. Mỗi hóa đơn (INVOICE) có mã số của nhà cung cấp (Vendor_ID), số của hóa đơn (Invoice_number), ngày phát hành (Invoice_date), tiền trả (Paid). => INVOICE là một thực thể diễn tả sự kiện nhận hàng về kho.
- ➔ Mỗi mặt hàng trong kho (INVENTORY ITEM) có mã số hàng (Item_number), đặc tả (Item_description), số lượng tồn (Quatity_in_stock), loại (Type_of_Item) và số lượng tối thiểu để đặt hàng lại (Minimum_order_quantity). => INVENTORY là thực thể mô tả kho.
- ➔ Mỗi hóa đơn đều có kèm danh mục các mặt hàng được giao (INVOICE ITEMS) chỉ ra rằng nhà cung cấp đã giao thêm một số mặt hàng nằm trong danh sách các mặt hàng của kho (IVENTORY ITEMS). Mỗi mặt hàng được giao có kèm theo số lượng giao (Quatity_added). => INVOICE ITEMS là một quan hệ giữa INVOICE và INVENTORY ITEMS, thuộc tính 'Quatity_added' là thuộc tính của quan hệ này.

- ➔ Các mặt hàng trong kho sẽ được sử dụng như nguyên liệu để làm ra sản phẩm (PRODUCT). Mỗi sản phẩm có mã số sản phẩm (Product_ID) và mô tả sản phẩm (Product_description).
- ➔ Vì mỗi sản phẩm được làm từ nhiều loại nguyên liệu khác nhau nên nhà hàng duy trì các công thức nấu ăn (RECIPE) để xác định sản phẩm nào cần (những) nguyên liệu gì và khối lượng bao nhiêu.
=> công thức nấu ăn (RECIPE) là một quan hệ giữa PRODUCT và INVENTORY ITEM. Liều lượng cần sử dụng (Quantity_Used) của mỗi loại nguyên liệu trong công thức là một thuộc tính của quan hệ RECIPE.
- ➔ Mỗi lần bán hàng là một SALE có số biên nhận (Receipt_number), và ngày bán (Sale_date). SALE là một thực thể mô tả sự kiện bán hàng.
- ➔ Khách hàng có thể mua nhiều sản phẩm cùng một lúc; do đó mỗi lần bán hàng sẽ có nhiều món hàng (sản phẩm) được liệt kê thành danh mục trong hóa đơn bán hàng (ITEM SALE), mỗi mục có số lượng bán (Quantity_sold). => ITEM SALE là một quan hệ giữa thực thể SALE và thực thể PRODUCT, trong đó Quantity_sold là thuộc tính của quan hệ này.

ERD cho hệ thống quản lý kho của Hoosie burger 51







- ➔ **Forms** : khuôn mẫu nhận dữ liệu vào hệ thống.
- ➔ **Reports** : khuôn mẫu thông tin cho người sử dụng.
- ➔ **Xác định thông tin nội bộ hoặc bên ngoài tổ chức**
Mẫu biểu (forms/reports) phải phù hợp với phạm vi sử dụng (bên trong/bên ngoài tổ chức).
- ➔ **Tài liệu xoay vòng**: Là tài liệu sử dụng cho cả nội bộ và bên ngoài, từ tổ chức ra khách hàng và quay về.
 - Phiếu bảo hành, thẻ tín dụng, hóa đơn định kỳ.
 - Barcode, vạch từ, vi mạch là phương tiện hỗ trợ tự động hóa đọc hoặc ghi dữ liệu cho các loại tài liệu này.

1. **Bảo vệ** hệ thống khỏi các tác tác động không mong muốn từ môi trường
2. **Lọc** bỏ các nội dung vào/ra không cần thiết
3. **Mã hóa** và **giải mã** các nội dung vào/ra
4. **Phát hiện lỗi** và **sửa lỗi**
5. **Lưu trữ tạm thời** thông tin, dữ liệu vào/ra bằng vùng đệm.
6. **Tổng hợp** và **chuyển đổi** dữ liệu sang khuôn mẫu cần thiết cho hệ thống (form) hoặc cho hệ thống khác (report).

1. Chuyển đổi từ ERD sang các bảng quan hệ (table) và các quan hệ giữa các bảng (relationship), để phù hợp với các hệ quản trị CSDL phổ biến hiện nay là loại Relational Database.
2. Thiết lập các cơ chế ràng buộc toàn vẹn dữ liệu để ngăn chặn các thao tác có thể gây sai sót trên CSDL
3. Chuẩn hóa các bảng quan hệ để bảo đảm toàn vẹn dữ liệu
4. Trộn các bảng quan hệ để tránh trùng lặp dữ liệu giữa các bảng.
5. Mã hóa hoặc nén dữ liệu lưu trữ , để giữ bí mật thông tin, tránh trùng lặp dữ liệu, hoặc giảm bớt không gian lưu trữ dữ liệu.
6. Tổ chức lưu trữ để thuận tiện truy xuất và sao lưu phục hồi khi có sự cố, hư hỏng trên CSDL

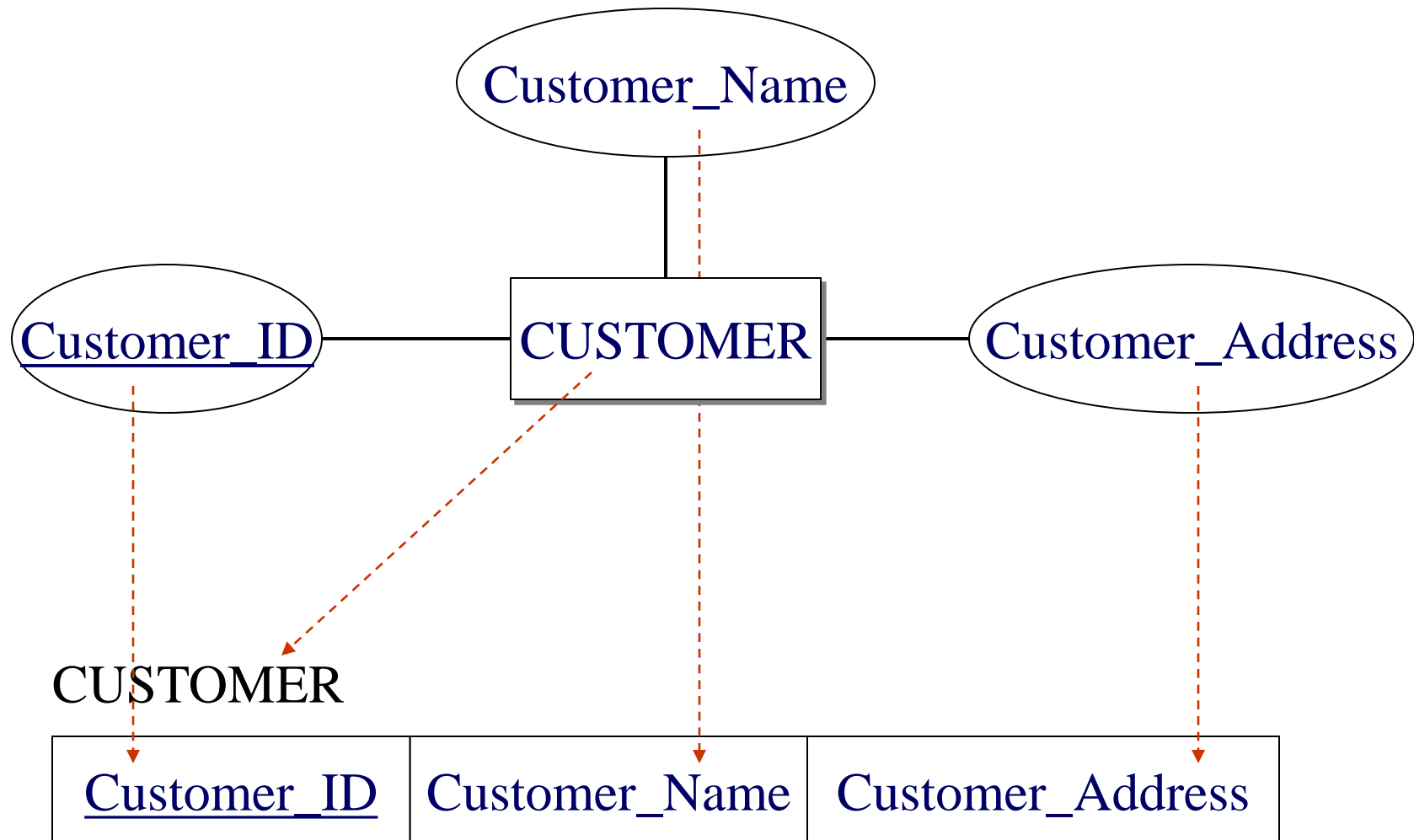
⇒ Là một cấu trúc bảng mô tả cho 1 thực thể

- Mỗi cột ứng với mỗi thuộc tính, gọi là **Field** (trường)
- Mỗi hàng ứng với mỗi thể hiện, gọi là **Record** (mẫu tin)
- Vd: EMPLOYEE2 (Emp_ID, Name, Dept_Name, Salary, Course_Title, Date_Complete). Emp_ID, Course_Title là một khóa chính kết hợp từ 2 thuộc tính của bảng quan hệ.

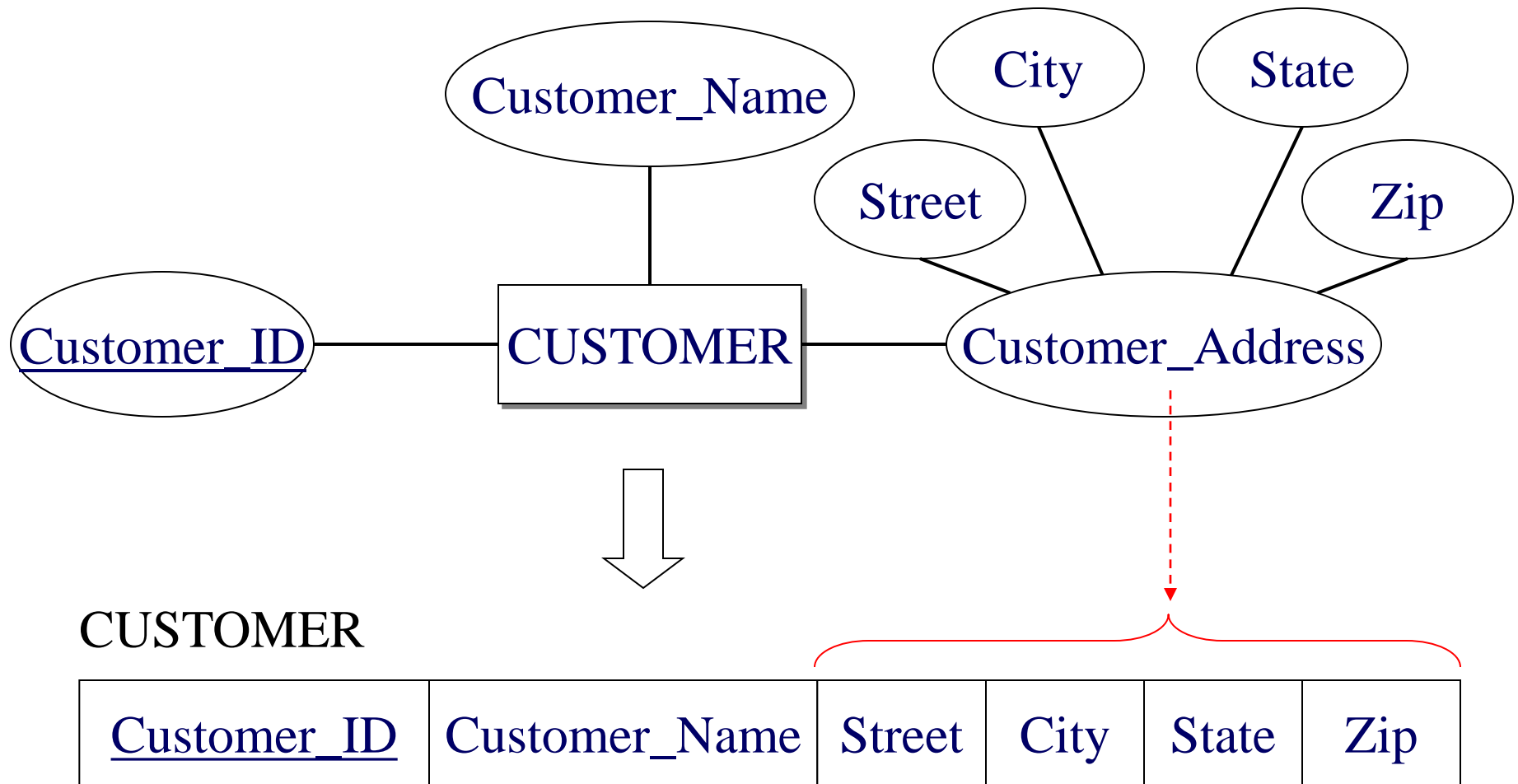
EMPLOYEE2

<u>Emp_ID</u>	Name	Dept_Name	Salary	<u>Course_Title</u>	Date_Completed
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/200X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/200X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/200X
110	Chris Lucero	Info Systems	43,000	SPSS	1/12/200X
110	Chris Lucero	Info Systems	43,000	C++	4/22/200X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/200X
150	Susan Martin	Marketing	42,000	Java	8/12/200X

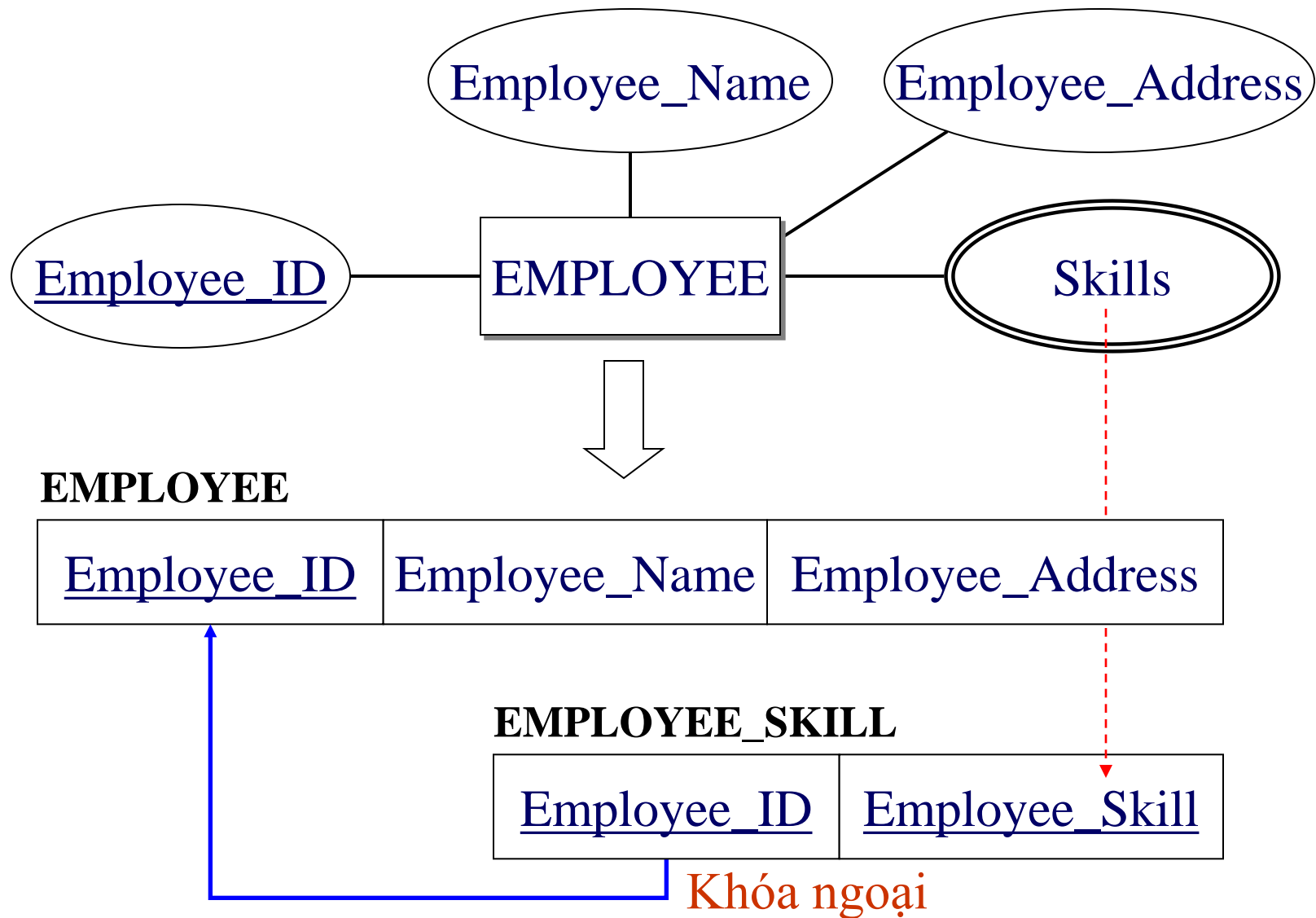
- ➡ Không phải tất cả các bảng dữ liệu đều là quan hệ; nó phải thoả mãn các điều kiện sau:
1. Mỗi bảng có 1 tên duy nhất
 2. Mỗi thuộc tính trong bảng có 1 tên duy nhất.
 3. Mỗi dòng là duy nhất. Không thể có 2 dòng có tất cả các cột đều giống nhau.
 4. Mỗi giá trị của thuộc tính là nguyên tố. Thuộc tính ‘composite’ là thuộc tính kết hợp nhiều thuộc tính. Vd:
Địa_chỉ = Số_nhà + Đường + Phường + Quận + Tp
- ➡ Các bảng quan hệ được tạo ra từ việc chuyển đổi lược đồ ERD sang bảng.

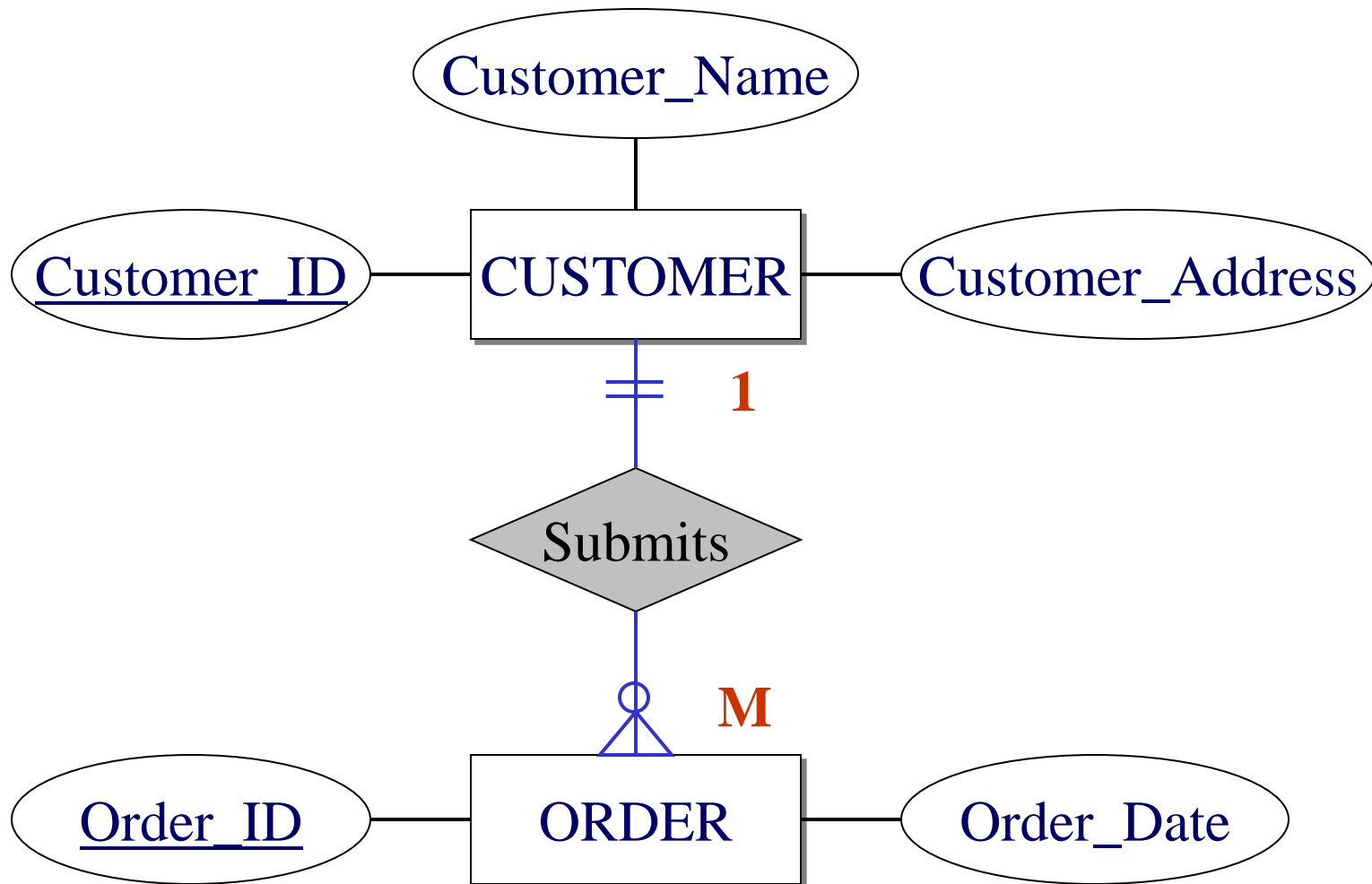


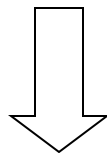
Thực thể sang bảng: 2. Composite attribute 59



Thực thể sang bảng: 3.Multivalue attribute 60







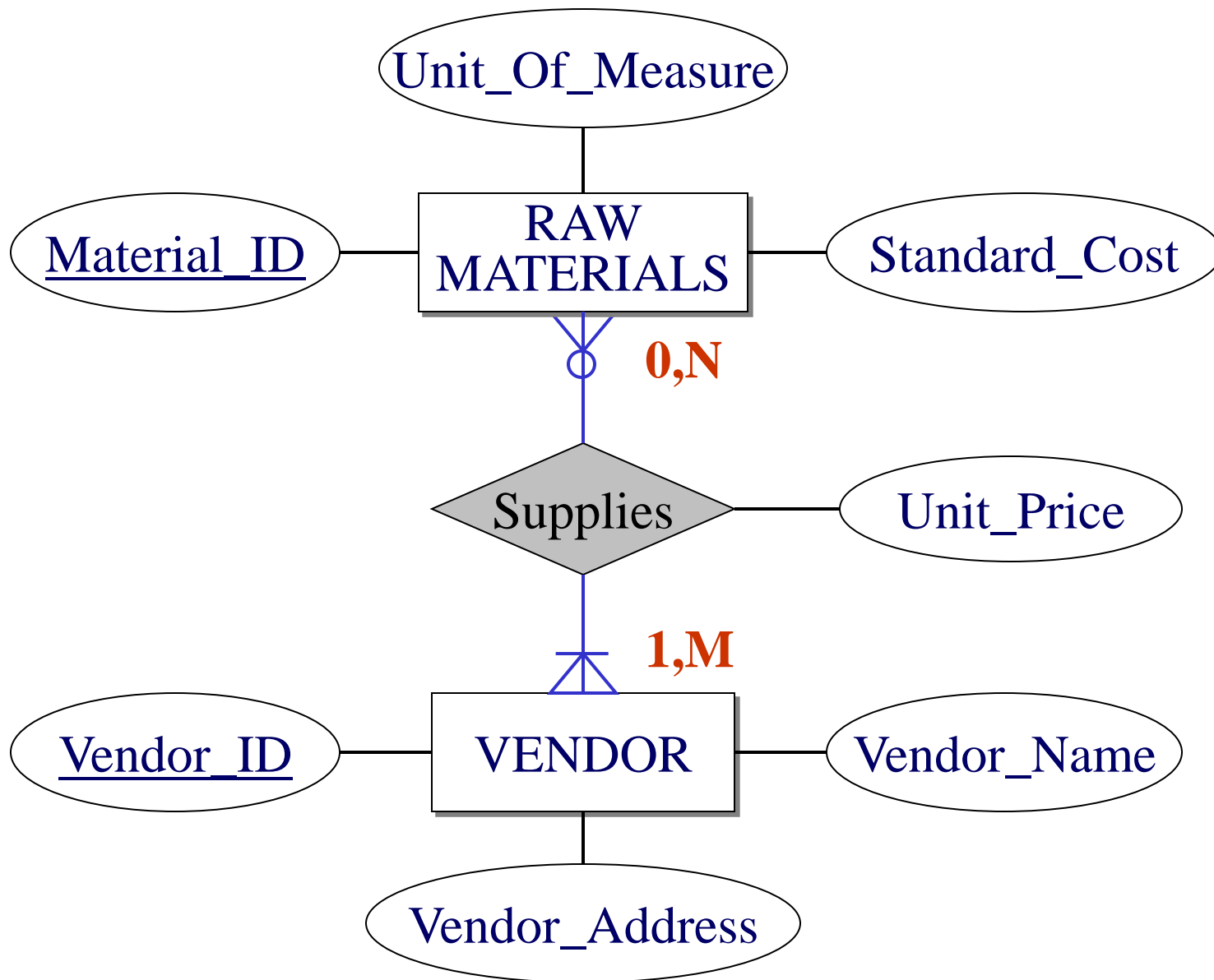
CUSTOMER

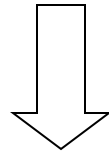
<u>Customer_ID</u>	Customer_Name	Customer_Address
--------------------	---------------	------------------

ORDER

<u>Order_ID</u>	<u>Customer_ID</u>	Order_Date
-----------------	--------------------	------------

Foreign key





RAW MATERIALS

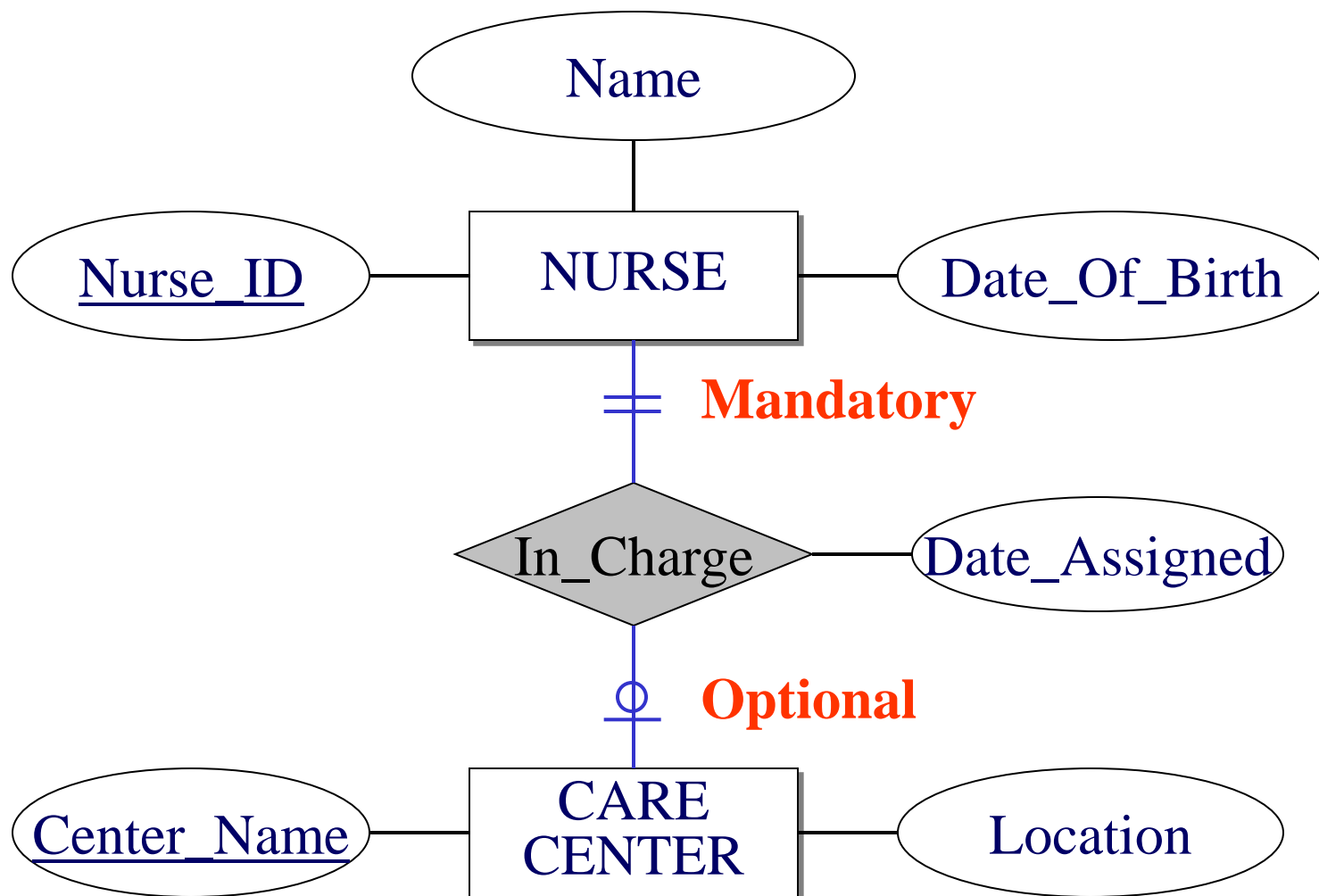
<u>Material_ID</u>	Standard_Cost	Unit_Of_Measure
--------------------	---------------	-----------------

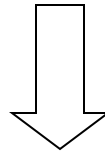
QUOTE

<u>Material_ID</u>	<u>Vendor_ID</u>	Unit_Price
--------------------	------------------	------------

VENDOR

<u>Vendor_ID</u>	Vendor_Name	Vendor_Address
------------------	-------------	----------------





NURSE

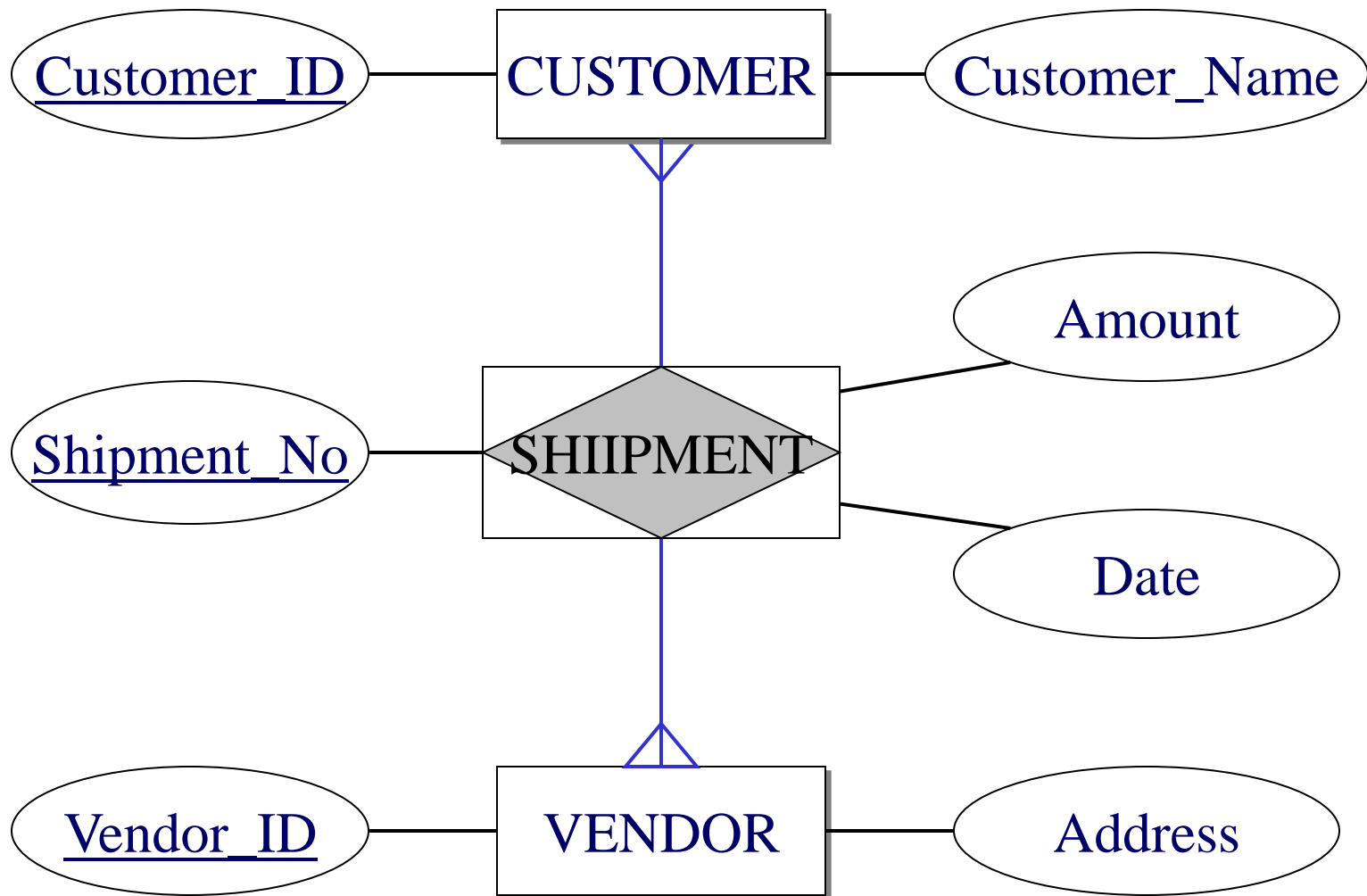
<u>Nurse_ID</u>	Name	Date_Of_Birth
-----------------	------	---------------

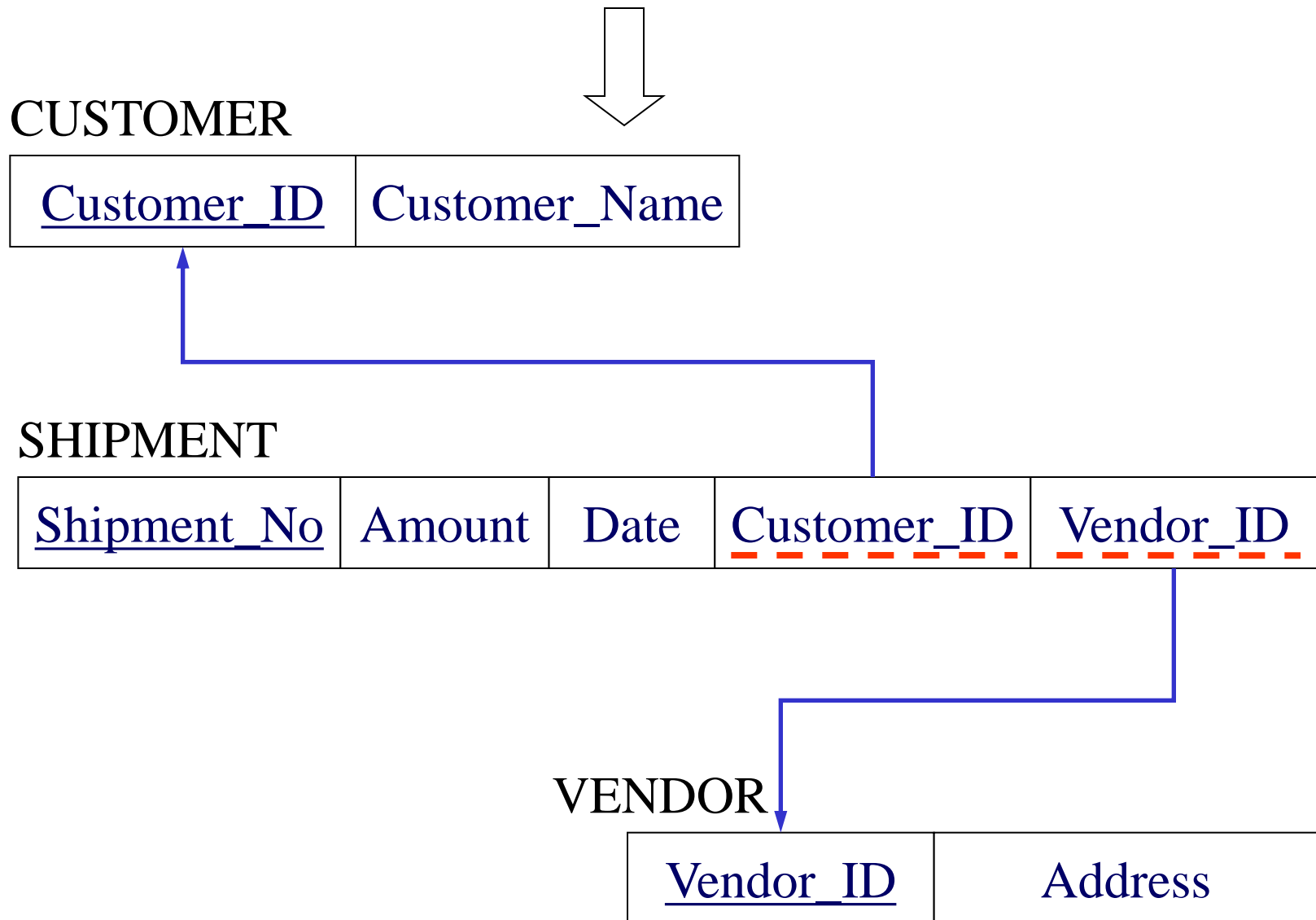
Mandatory 1

CARE CENTER

<u>Center_Name</u>	Location	<u>Nurse_In_Charge</u>	Date_Assigned
--------------------	----------	------------------------	---------------

Foreign key





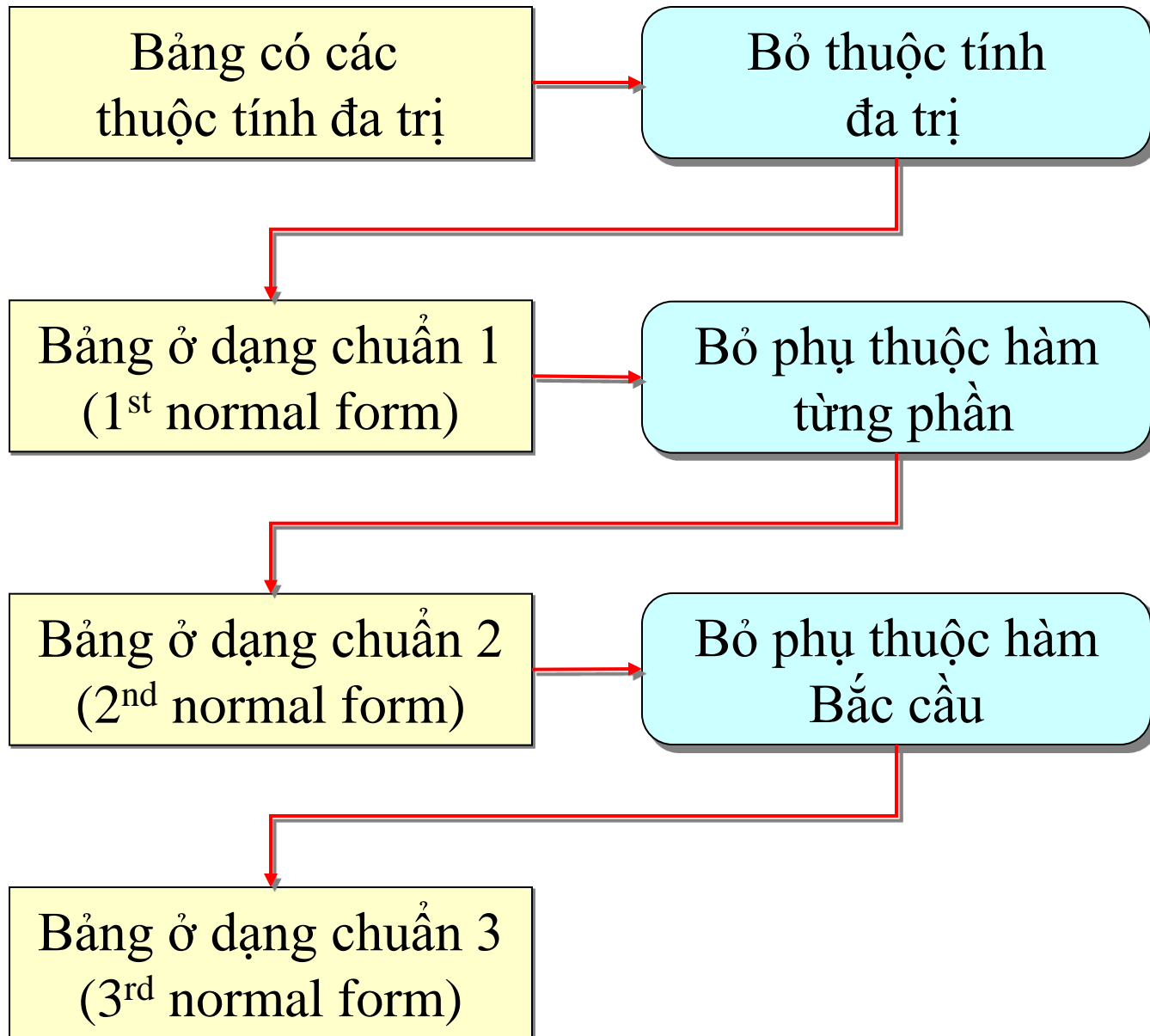
⇒ Ràng buộc trên miền giá trị của mỗi field

- Mỗi thuộc tính của thực thể được quy định một miền giá trị hợp lệ. Các giá trị nằm ngoài miền giá trị này là các giá trị vô nghĩa đối với thực thể, cần phải loại bỏ. Ví dụ: thuộc tính Nam/Nữ chỉ có 2 giá trị

⇒ Ràng buộc toàn vẹn thực thể (và bảo toàn các liên kết)

- Không có khóa chính nào có thể rỗng – tất cả các khóa chính phải có chứa dữ liệu để phân biệt các dòng (mẫu tin) trong bảng
- Bảo toàn liên kết khi thêm, xóa, sửa dữ liệu để không mất dữ liệu

⇒ Để tăng tính toàn vẹn dữ liệu, đồng thời các bảng quan hệ có cấu trúc tốt (chứa tối thiểu dữ liệu dư thừa + cấu trúc cho phép thiết lập các ràng buộc toàn vẹn thực thể), người ta cần phải **chuẩn hóa các bảng quan hệ**.



Bảng quan hệ dạng chuẩn 1 là bảng quan hệ không chứa nhiều giá trị trong một ô, hoặc không có nhiều cột đồng nghĩa.

➡ Đặc điểm: có nhiều dữ liệu bị trùng lặp giữa các dòng

1st-NF relation

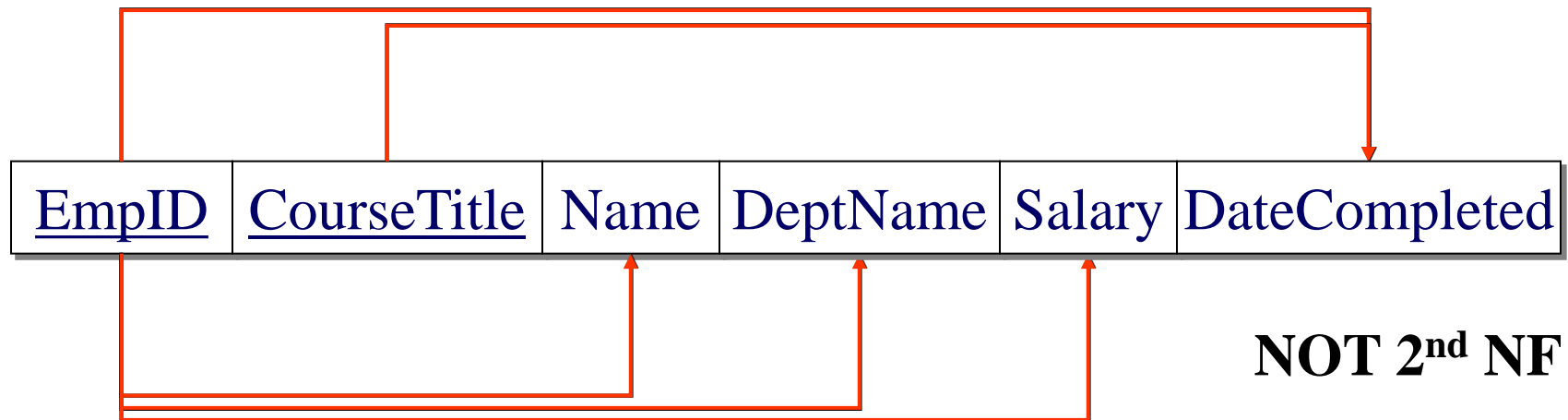
EMPLOYEE2

<u>Emp_ID</u>	Name	Dept_Name	Salary	<u>Course_Title</u>	Date_Completed
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/200X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/200X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/200X
110	Chris Lucero	Info Systems	43,000	SPSS	1/12/200X
110	Chris Lucero	Info Systems	43,000	C++	4/22/200X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/200X
150	Susan Martin	Marketing	42,000	Java	8/12/200X

Bảng dạng chuẩn 2 là bảng ở dạng chuẩn 1 và mọi thuộc tính không phải là khóa phụ thuộc hàm vào toàn bộ khóa chính

- Mỗi thuộc tính không khóa được xác định bởi toàn bộ khóa chính, không phải chỉ một phần của khóa chính
- Không có thuộc tính không khóa phụ thuộc vào 1 phần của khóa chính

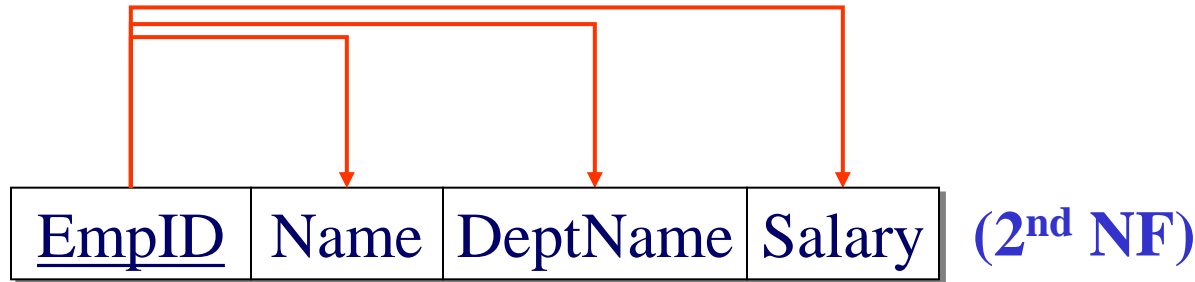
DateCompleted phụ thuộc hàm hoàn toàn vào (EmpID, CourseTitle)



NOT 2nd NF !!

Name, DeptName, Salary chỉ phụ thuộc vào (Emp_ID)

1. Các thuộc tính chỉ phụ thuộc vào 1 phần khóa chính là EmpID kết hợp với EmpID để tạo thành 1 bảng chuẩn 2



3. Đặt quan hệ giữa 2 bảng mới



2. Các thuộc tính phụ thuộc hoàn toàn vào khóa chính kết hợp với khóa chính để tạo thành 1 bảng chuẩn 2

Bảng dạng chuẩn 3 là bảng dạng chuẩn 2 và không chứa phụ thuộc hàm bắc cầu . Phụ thuộc hàm bắc cầu: một thuộc tính C phụ thuộc hàm vào thuộc tính B, thuộc tính B lại phụ thuộc hàm vào thuộc tính A, ký hiệu là $A \rightarrow B \rightarrow C$).



Cust_ID \rightarrow Salesperson \rightarrow Region : phụ thuộc bắc cầu

SALES

Cust_ID	Name	Salesperson	Region
8023	Anderson	Smith	South
9167	Bancroft	Hicks	West
7924	Hobbs	Smith	South
6837	Tucker	Hernandez	East
8596	Eckersley	Hicks	West
7018	Arnold	Faulb	North

Biến đổi thành dạng chuẩn 3

75

(bảng không chứa thuộc tính phụ thuộc bắc cầu 'Region')

SALES1

Cust_ID	Name	Salesperson
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

(bảng chứa thuộc tính phụ thuộc bắc cầu 'Region' và khóa của nó)

SPERSON

Salesperson	Region
Smith	South
Hicks	West
Hernandez	East
Faulb	North

Salesperson → Region

Cust_ID → Name, Salesperson

SPERSON

<u>Salesperson</u>	Region
--------------------	--------

SALES1

<u>Cust_ID</u>	Name	Salesperson
----------------	------	-------------

➔ Sau khi thực hiện normalization, một số bảng quan hệ có thể dư thừa vì cùng mô tả cùng một hoặc một số thực thể tương tự nhau, có thể trộn (merge) thành 1 bảng

EMPLOYEE1(Emp_ID, Name, Address, Phone)

EMPLOYEE2(Emp_ID, Name, Address, Jobcode,
Number_of_years)

- Sau khi trộn, ta được:

EMPLOYEE(Emp_ID, Name, Address, Phone, Jobcode,
Number_of_years)

➔ Việc trộn các bảng phải bảo toàn ý nghĩa của dữ liệu

1. Trường hợp đồng nghĩa

2. Trường hợp đồng âm, dị nghĩa

3. Có thể sinh ra phụ thuộc hàm bắc cầu, cần loại bỏ

Ví dụ: 1.Đồng nghĩa

STUDENT1(Stu_ID, Name)

STUDENT2(Matriculation_number, Name, Address)

Nếu Stu_ID và Matriculation_number cùng được dùng để diễn tả cùng một nội dung là số an sinh xã hội (Social Security Number), hai bảng này có thể trộn lại và sử dụng khóa mới là SSN :

STUDENT (SSN, Name, Address)

Ví dụ: 2.Đồng âm, dị nghĩa

STUDENT1(Stu_ID, Name, Address)

- “Address”: địa chỉ của SV **trong khuôn viên trường**

STUDENT2(Stu_ID, Name, PhoneNumber, Address)

- “Address”: địa chỉ **nhà riêng** của SV

Vì Address ở 2 bảng mô tả cho 2 nội dung khác nhau, nên khi trộn 2 bảng, 2 thuộc tính này cần đặt lại 2 tên khác nhau để phân biệt:

STUDENT (Stu_ID, Name, Campus_Address,
Permanent_Address)

Ví dụ: 3. Phụ thuộc hàm bậc cao

STUDENT1 (Stu_ID, Advisor) là bảng 3rd NF

STUDENT2 (Stu_ID, Major) là bảng 3rd NF

⇒ Sau khi trộn 2 bảng, sẽ có bảng mới là

STUDENT (Stu_ID, Advisor, Major)

⇒ Tuy nhiên, nếu mỗi người hướng dẫn chỉ hướng dẫn 1 chuyên ngành (có phụ thuộc hàm **Advisor -> Major**) thì bảng trên không là bảng chuẩn 3rd NF, cần phải bỏ phụ thuộc hàm bậc cao một lần nữa để tạo ra 2 bảng:

STUDENT (Stu_ID, Advisor)



ADVISOR-MAJOR (Advisor, Major)

BILLING_ADDRESS

Cust_No	Block_no	Street	City_State
1273	2226	Plainfield Ave	NJ
6390	2110	Plainfield Ave	NJ

← 40 bytes →

BILLING_ADDRESS

Cust_No	Block_no	Ref	City_State
1273	2226	1001	NJ
6390	2110	1001	NJ

← 4 bytes →

Range: 0 .. 9999

Ref	Street
1001	Plainfield Ave
7024	Oak

LOOKUP_STREET

Tính khả thi của các cải tiến được xem xét trên các yếu tố sau:

- ➔ Năng lực của hệ thống mới đối với các kế hoạch phát triển tổ chức, ie, xem xét khả năng thỏa mãn các tiêu chí đánh giá mức độ thành công (CSFs) của tổ chức.
- ➔ Phạm vi thay đổi của hệ thống thông tin mới trong tổ chức, được diễn tả theo mức độ từ thấp đến cao tương ứng với mức độ rủi ro và thành quả từ thấp đến cao như sau:
 1. Tự động hóa (thay thế xử lý nhân công)
 2. Hợp lý hóa (sắp xếp tối ưu công việc trên dây chuyền)
 3. Tái cấu trúc tiến trình kinh doanh (giữ nguyên mục tiêu, thay đổi phương pháp, cấu trúc tổ chức, nguồn lực)
 4. Chuyển dịch cơ cấu tổ chức (thay đổi mục tiêu kinh doanh).

1. Giả sử ta có bảng quan hệ $R(\underline{A}, \underline{B}, C, D, E)$ với khóa chính A, B và phụ thuộc hàm $A \rightarrow C$ và $D \rightarrow E$. Hãy chuyển quan hệ sang dạng chuẩn 3NF.
2. Quan hệ MEETING có các thuộc tính và khóa như sau:
 $MEETING(\underline{CourseName}, CourseTitle, \underline{Day}, \underline{Time}, BuildingName, Room\#, Address)$. Giả sử có 2 phụ thuộc hàm trên quan hệ này:
 - $Address$ phụ thuộc hàm vào $BuildingName$
 - $CourseTitle$ phụ thuộc hàm vào $CourseName$Hãy chuyển quan hệ sang dạng 3NF.
3. Quan hệ BUILDING có các thuộc tính và khóa như sau:
 $BUILDING(\underline{Street}, \underline{Street\#}, \underline{City}, OwnerName, OwnerAddress, PropertyTax)$. Giả sử có 3 phụ thuộc hàm trên quan hệ này:
 1. $OwnerName$ phụ thuộc hàm vào $street, street\#, city$
 2. $PropertyTax$ phụ thuộc hàm vào $city$
 3. $OwnerAddress$ phụ thuộc hàm vào $ownerName$Hãy chuyển quan hệ sang dạng 3NF.

4. Một hãng xe hơi có CSDL lưu các thông tin sau:

➡ Thông tin về nhà cung cấp (Suppliers)

Mỗi nhà cung cấp được hãng xe hơi gán một số định danh Supplier# để phân biệt nhau

Mỗi nhà cung cấp có tên (SupplierName), và thành phố (SupplierCity)

Mỗi nhà cung cấp cung cấp 1 hoặc nhiều phụ tùng cho xe

➡ Thông tin về phụ tùng xe (Parts)

Mỗi phụ tùng xe có một tên (PartName), số định danh (Part#) và giá (PartPrice)

Mỗi phụ tùng được cung cấp bởi 1 hay nhiều nhà cung cấp

Part# là số dùng để phân biệt các phụ tùng xe với nhau

➡ Thông tin về đợt cung cấp (Supply)

Mỗi đợt cung cấp có nhà cung cấp cung cấp một phụ tùng

Mỗi đợt cung cấp có số lượng(quantity), ngày (date), tên bộ phận (partName), thành phố của nhà cung cấp (SupplierCity), và mã vùng của nhà cung cấp (Supplier_Postal)

Giả sử hãng đã biết được các quan hệ phụ thuộc như sau:

1. Số lượng trong mỗi đợt cung cấp hàng phụ thuộc hoàn toàn vào supplier#,part#,date
2. PartName được xác định bởi Part#
3. SupplierCity,SupplierPostal phụ thuộc vào Supplier#
4. Nếu biết SupplierPostal, ta sẽ biết được SupplierCity

a) Hãy vẽ lược đồ ERD cho CSDL.

b) Chuyển lược đồ ERD sang bảng quan hệ và chuẩn hóa thành dạng 3NF.

XIN CẢM ƠN!